Raport dla Klubu Rzymskiego:
Wpływ mikroelektroniki na zatrudnienie

P. 1877/83

5 1983

informatyka

Standard BASICA

Rastrowe systemy graficzne

MOTOROLA 68000

ZX81

WYDAWNICTWO NOT SIGMA

Ułatwienia

Docierające do nas ostatnio opinie Czytelników o INFORMATYCE są najczęściej pozytywne, coraz rzadziej słyszymy istotne zastrzeżenia. Przy tym ubiegłoroczny katastrofizm, zapowiadający upadek polskiej informatyki, zaczyna oddawać prym racjonalnej kalkulacji, a w konsekwencji — wizji bardziej optymistycznej. To także podnosi wagę pisma.

Ale nasze kłopoty nie mijają. Pisaliśmy o nich w tegorocznym drugim numerze — spada nakład, INFORMATYKA zaczęła przynosić deficyt.

Można zatem wyciągnąć wniosek, że wprawdzie pismo jest pożyteczne dla coraz większej grupy odbiorców, niemniej jego wysoka cena i nie dość precyzyjnie — z konieczności — określony profil zniechęcają do zamawiania prenumeraty. A ta forma rozpowszechniania, jak wskazuje praktyka, jest jedyną drogą regularnego otrzymywania kolejnych numerów.

Dla instytucji zainteresowanych INFORMATYKĄ, ważną informacją będzie stwierdzenie Ministra Finansów, który rozwiewając wątpliwości Naczelnej Organizacji Technicznej wyjaśnił, że zakłady pracy mają prawo dokonywać zakupu prasy technicznej ze środków funduszu postępu techniczno-ekonomicznego (szczegóły na str. 7). Może dzięki temu nie trzeba będzie rezygnować z prasy specjalistycznej, której oszczędnościowy niedobór (jak mówi fachowe wyliczenie: 1 egzemplarz powinien przypadać na ok. 10 pracowników) wydaje się być wątpliwym sposobem obrony przed kryzysem.

Aby natomiast zachęcić prenumeratorów indywidualnych, Wydawnictwo SIGMA ogłosiło wprowadzenie od 1 lipca br. prenumeraty ulgowej dla członków stowarzyszeń naukowo-technicznych NOT, studentów i uczniów szkół zawodowych. Za INFORMATYKĘ będą oni płacić 50 zł.

Dla pozostałych nabywców nie mamy żadnych przyjemnych wiadomości. Poza powtórzeniem deklaracji, że INFORMATYKA będzie coraz ciekawsza — na miarę jej (informatyki) światowego rozwoju.

Redakcja

NACZELNA ORGANIZACJA TECHNICZNA

WYDAWNICTWO SIGMA

CZASOPISM I KSIĄŻEK TECHNICZNYCH

ul. Świętokrzyska 14a
00-950 Warszawa
skrytka pocztowa 1004

KOLEGIUM REDAKCYJNE

Redaktor naczelny: prof. dr hab. Leon ŁUKASZEWICZ

mgr inż. Zbigniew GLUZA, dr Janusz GWIAZDA, Władysław KLEPACZ (zastępca redaktora naczelnego), dr inż. Tomasz PAWLAK, mgr Andrzej SZAŁAS, mgr Zbigniew ŚWIRSKI, dr inż. Janusz ZALEWSKI

Sekretarz redakcji: mgr Teresa JABŁOŃSKA

RADA PROGRAMOWA

Prof. dr hab. Tadeusz PECHE (przewodniczący), mgr inż. Tomasz BAŃKOWSKI (sekretarz), mgr inż. Antoni BOSSOWSKI, mgr inż. Roman BURNO, prof. dr hab. Andrzej JANICKI, mgr inż. Jan KRAMARCZUK, prof. dr hab. inż. Juliusz KULIKOWSKI, prof. dr hab. Leon ŁUKASZEWICZ, prof. dr hab. Antoni MAZURKIEWICZ, gen. dr inż. Marian PASTERNAK, dr inż. Bronisław PIWOWAR, mgr Zbigniew SUBSTYK, prof. dr hab. Tadeusz WALCZAK

Materiałów nie zamówionych Redakcja nie zwraca.

Redakcja: 00-041 Warszawa, ul. Jasna 14/16, pok. 243 i 244, tel. 27-71-40, dyżury redakcji 10.00—12.00

Zakł. Graf. „Tamka". Zam. 2121. Obj. 5,0 ark. druk. Nakład 3550 egz. M-95

Cena egzemplarza zł 75.— INDEKS 36124 Prenumerata roczna zł 900.—

P. 1877/83

Informatyka

zastosowania w gospodarce, technice i nauce

Nr 5
MIESIĘCZNIK
1983
ROK XVIII
Maj

ORGAN KOMITETU INFORMATYKI MINISTERSTWA NAUKI, SZKOLNICTWA WYŻSZEGO I TECHNIKI ORAZ KOMITETU NAUKOWO-TECHNICZNEGO NOT DS. INFORMATYKI

W NUMERZE:

Strona

Raport dla Klubu Rzymskiego — Wpływ komputeryzacji na miejsce pracy
John Evans 2

Standaryzacja języka BASIC (1). Norma języka minimalnego
Wacław Iszkowski, Marek Maniecki 8

Klasy w LOGLANIE
Hanna Oktaba 12

Parametry użytkowe wybranych rastrowych systemów graficznych
Jacek Owczarczyk 15

Przegląd mikroprocesorów 16-bitowych (3). MOTOROLA 68000
Jerzy Grabowski 18

SYSTEMY

Komputer osobisty ZX81
Jan Dawidowski 21

Rozszerzenie pakietu informacyjno-wyszukiwawczego SIM-400M
Marian Furman, Konrad Karpiński, Zbigniew Nowicki, Jerzy Solak 24

Translator języka adresów bezwzględnych dla procesora autonomicznego CAMAC typu 131
Zbigniew Rempała, Roman Rygał 27

ALGORYTMY

Dwa algorytmy probabilistyczne
Andrzej Szałas, Zbigniew Świrski 28

Z KRAJU

Zastosowanie sterownika mikroprocesorowego MIKRO-80 w gospodarce
Janusz Zalewski 30

Klub Użytkowników Mikroprocesorów
Andrzej Drożniak 31

GIEŁDA INFORMACJI 31

ZE ŚWIATA

Gdy komputer będzie słyszał
Oprac. Marianna Sobczyk 32

Seria HP 9000 *(MaS)* 33

RECENZJE

Niekonieczna „Praktyka programowania”
Marek Sobczyk 35

„Tablice rozproszone”
Andrzej Szałas 35

TERMINOLOGIA

Kwestie terminologiczne
Andrzej Kamieński 37

GRY

Firma
Z. Świrski 37

W SKRÓCIE 38

POGLĄDY

Moje podglądy
Janusz Zalewski III okł.

Raport dla Klubu Rzymskiego: Mikroelektronika i społeczeństwo
Na dobre i złe (1)

W poprzednim numerze zamieściliśmy recenzję Raportu dla Klubu Rzymskiego „Mikroelektronika i społeczeństwo. Na dobre i złe". W Raporcie tym opisano wizję społeczeństwa, jaka wyłania się z analizy dzisiejszego tempa rozwoju informatyki opartej na mikroukładach. Raport przedstawia bowiem procesy, którym ulega coraz bardziej skomputeryzowana ludzkość.

Poniżej rozpoczynamy publikację wybranych fragmentów Raportu. Spośród 11 składających się nań rozdziałów-esejów wybraliśmy dwa, wyraźnie — naszym zdaniem — ujmujące przemiany społeczne, których nie tylko narzędziem, ale i motorem staje się informatyka. Poza esejem Johna Evansa (podzielonym na dwie części), przedstawimy wypowiedź Franka Barnaby o zastosowaniach mikroelektroniki w urządzeniach wojskowych.

Wybrane fragmenty nie wyczerpują oczywiście tematu. Zresztą, także sam Raport go nie wyczerpuje. W przyszłości spróbujemy pogłębić zarysowany przezeń obraz — tak, by informatycy byli świadomi ogólniejszych następstw rozwijanej przez siebie technologii.

Pięć esejów Raportu opublikował PRZEGLĄD TECHNICZNY, całość przygotowywana jest do druku w Wydawnictwie „Książka i Wiedza".

Redakcja

JOHN EVANS
Europejski Instytut Związków Zawodowych
Bruksela

Wpływ komputeryzacji na miejsce pracy

Pierwsze pytanie, jakie nasuwa się w związku z mikroelektroniką, dotyczy tego, czy wprowadzenie nowej technologii nie spowoduje zmniejszenia liczby miejsc pracy. W jednym z rozdziałów Raportu, zatytułowanym „Mikroelektronika a makroekonomia", przedstawiono skalę przesunięć w zakresie miejsc pracy, jakie w ciągu najbliższych dziesięciu lat zapewne nastąpią — na skutek wprowadzenia nowych technologii opartych na mikroelektronice. Równie ważny jest jednak wpływ mikroelektroniki na charakter ludzkiej pracy oraz na środowisko, w jakim ją się wykonuje.

Dzięki zastosowaniu mikroelektroniki w wyrobach i procesach, robotnicy zatrudnieni w takich gałęziach przemysłu, jak mechanika precyzyjna, technika motoryzacyjna, drukarstwo, a także w niektórych usługach (np. w bankowości i ubezpieczeniach) — już odczuwają pewne istotne zmiany. Wpływ tych zmian na jakość pracy jest bardzo złożony. Niektóre stanowiska zostają wyeliminowane, inne wymagają przekwalifikowania pracownika, jeszcze inne — wyższych niż poprzednio kwalifikacji. W niektórych przypadkach, dzięki zastosowaniu mikroelektroniki polepszają się fizyczne warunki pracy: w innych natomiast, z tej samej przyczyny, występują problemy związane z rosnącą izolacją pracownika, zaostrzeniem nadzoru oraz odmiennym rytmem pracy.

Te sprzeczne skutki świadczą o tym, że z samej technologii opartej na mikroelektronice bynajmniej nie wynika jakaś szczególna organizacja pracy lub jakiś określony zespół warunków pracy. To raczej cele ekonomiczne, techniczne i społeczne, które dana technologia pozwala osiągnąć, determinują jej wpływ na warunki pracy. Biorąc pod uwagę długofalowe perspektywy dotyczące możliwości gospodarczych i społecznych, otwierających się dzięki nowej technologii, można stwierdzić, że istnieją pewne realne szanse na polepszenie warunków pracy, dzięki powszechnemu zastosowaniu mikroelektroniki. Jeżeli jednak za zmianą technologii kryje się jedynie pęd do zwiększania produktywności i osiągania szybkich zysków, to w rezultacie może dojść do ogólnego pogorszenia jakości pracy. Najczęściej właśnie taki był charakter zmian technologicznych wprowadzanych dotychczas; gdziekolwiek brano pod uwagę czynniki społeczne, wynikało to zazwyczaj z przyczyn ekonomicznych, gdy występowały trudności z zatrudnieniem z powodu nieatrakcyjnych warunków pracy.

Istnieje niebezpieczeństwo, że na tle obecnego kryzysu gospodarczego i znacznego bezrobocia będzie dążyć się za wszelką cenę do racjonalizacji wynikającej jedynie ze zmiany technologii. W konsekwencji, wzbierająca fala zmian technologicznych opartych na mikroelektronice niesie niebezpieczeństwo pominięcia i zaprzepaszczenia szeregu innych możliwości.

JAKOŚĆ PRACY

Określanie i mierzenie aspektów jakościowych pracy jest procesem bardzo subiektywnym. W praktyce jednak każdy człowiek ocenia swą pracę uwzględniając aspekty przy-

noszące mu satysfakcję lub wzbudzające jego zastrzeżenia czy sprzeciw. Ocenia: treść lub istotę pracy, poczucie sensu i wartość poznawczą pracy, środowisko i bezpieczeństwo pracy, kontakty społeczne, wynagrodzenie oraz wpływ pracy na czas i charakter wypoczynku. Do najważniejszych czynników, od których zależy istota wykonywanego zawodu należą m.in. kwalifikacje, odpowiedzialność, niezależność i samodzielność w ustalaniu tempa pracy. Przy określaniu sensu czy znaczenia pracy, ważne jest, by pracownik rozumiał, jaką rolę w całym procesie produkcyjnym odgrywa on sam oraz jaki związek ma jego praca z produktem końcowym.

Zastosowanie mikroelektroniki wpływa na wszystkie aspekty pracy niezależnie od tego, czy rzutuje ono na zmianę procesu produkcji, czy przejawia się w formie usług niezbędnych w tym procesie, czy też wpływa na zmianę produktu końcowego. W rozdziale niniejszym zastanowimy się, w jaki sposób mikroelektronika wpływa na jakościowe cechy pracy, a następnie przejdziemy do rozważań nad możliwościami technologicznymi wynikającymi z jej zastosowań.

MIKROELEKTRONIKA A KWALIFIKACJE

Początkowe rozdziały Raportu poświęcono omówieniu potencjalnych możliwości, które wynikają z zastosowania mikroelektroniki w produktach i procesach. Na przykładzie niektórych zastosowań widać szereg pozytywnych i negatywnych konsekwencji mikroelektroniki dla kwalifikacji zawodowych pracowników. Z jednej strony — pewna liczba stanowisk pracy ulegnie całkowitej likwidacji, zaś wymagania dotyczące innych stanowisk zmniejszą się lub zmienią. Z drugiej strony — w niektórych rodzajach pracy wymagania kwalifikacyjne będą znacznie większe. Te trzy efekty, czyli likwidacja niektórych stanowisk pracy, konieczność zmiany kwalifikacji lub ich podwyższenia, składają się na proces obecnie zachodzących zmian.

Mikroelektronika umożliwi wyeliminowanie wielu ciężkich i monotonnych zajęć, inne zaś wymagające dotąd wysokich kwalifikacji, ulegną standaryzacji i uproszczeniu. Standaryzacja prowadzi do skupienia wymagań dotyczących umiejętności na niewielu nowych stanowiskach pracy, utworzonych przede wszystkim w sferze planowania i usług produkcyjnych. W tych nowych dziedzinach praca polega często na posługiwaniu się komputerem.

Likwidacja stanowisk pracy nie wymagających kwalifikacji

Badania wykazały szczególną skuteczność zastosowań mikroelektroniki w automatyzacji procesów przemysłowych, w których wykonywane są powtarzalne czynności. W rozwoju społeczeństw przemysłowych regułą było tworzenie takich rodzajów pracy, w których można było zatrudniać osoby z niskimi kwalifikacjami. Wynikało to z podziału pracy, specjalizacji zawodowej i mechanizacji procesów produkcyjnych.

Nie był to wszakże prosty rezultat mechanizacji przez zmianę technologii, lecz raczej efekt zastosowania teorii naukowych lub zasady tayloryzmu. Systemy oparte na tych teoriach cechowało rozdzielenie planowania i wykonania pracy, rozbicie pracy na najdrobniejsze i najprostsze czynności oraz sztywna i surowa kontrola kierownicza. System linii produkcyjnej przyjął się i upowszechnił w znacznej części wytwórczości przemysłowej, gdzie w grę wchodziła produkcja masowa. Ostatnio jednak coraz powszechniejsze stają się alternatywne sposoby produkcji, nie oparte na liniowej ciągłości procesów.

Z badań przeprowadzonych w Niemczech Zachodnich wynika, że w 1977 roku, na ogólną liczbę zatrudnionych — 33% mężczyzn i 46% kobiet wykonywało powtarzalne, monotonne zadania, nie wymagające dużej odpowiedzialności. Stwierdzono jednocześnie, że 11% mężczyzn i 40% kobiet spełniało w pracy proste funkcje pomocnicze. Dzięki mikroelektronice istnieje poważna szansa pełnej automatyzacji takich właśnie rodzajów pracy.

Przykład 1. Roboty przemysłowe

Jedno z zastosowań mikroelektroniki obejmuje wprowadzane coraz powszechniej roboty, dzięki którym działania człowieka zastępowane są wprost przez działania mechaniczne. Zakres zastosowań oraz rozwój konstrukcji robotów omówiono w innych rozdziałach Raportu. Dotychczas, roboty stosowano głównie w przemysłach nastawionych na produkcję masową, jak np. przemysł motoryzacyjny (roboty pracują już w większości amerykańskich, europejskich i japońskich fabryk samochodów), obróbka metali, przemysł chemiczny, inżynieria lądowa i wodna, przemysł elektryczny i elektroniczny, ale zapewne w ciągu najbliższych dziesięciu lat znajdą one znacznie szersze obszary zastosowań.

Roboty wykorzystuje się głównie do montowania części, ich łączenia i manipulowania nimi. Według jednego z czołowych amerykańskich producentów robotów, 50% maszyn używa się do zgrzewania punktowego, blisko 11% do odlewania kokilowego i 5% do zmechanizowanego załadunku. W miarę rozwoju zastosowań mikroelektroniki, zwłaszcza w nauce i technice, powstają nowe konstrukcje maszyn uniwersalnych, zdolnych do wykonywania coraz bardziej skomplikowanych czynności montażowych.

Badania na temat wprowadzania robotów w RFN, omówione w rozdziale pt. „Wpływ na przedsiębiorstwo", dotyczyły praktycznego zastosowania dziesięciu robotów do spawania łukowego, malowania natryskowego i manipulowania częściami — w pięciu firmach reprezentujących różne sektory przemysłu. W poniższej tabeli przedstawiono przesunięcia i zmiany personelu, jakie trzeba było wprowadzić. Objęły one łącznie 46 osób wykonujących prace nie wymagające wysokich kwalifikacji. Okazało się, że 7 osób jest zbytecznych, 28 przeniesiono do innego sektora, 10 można było przesunąć do innej pracy w tym samym sektorze, a jedną osobę trzeba było zatrudnić. Użycie robotów wprowadziło zatem istotną zmianę w zakresie prac wykonywanych przez załogę.

Tabela. Zmiany personalne w wyniku zastosowania robotów przemysłowych (na podstawie badań dotyczących wprowadzenia dziesięciu robotów w pięciu zachodnioniemieckich zakładach przemysłowych)

Kategoria personelu	Liczba osób, którym zmieniono obowiązki pracy					
	Robotnicy usunięci z sektora	Robotnicy przyjęci do sektora	Zmiana personelu w ramach sektora	Robotnicy zwolnieni z pracy	Nabór nowych kadr	Razem
Personel wykwalifikowany	4	6	2	—	—	12
Personel niewykwalifikowany	28	1	10	7	—	46
Razem	32	7	12	7	—	58

Źródło: Battelle, Genewa

Inne badania, dotyczące czterdziestu przypadków zastosowania robotów w jednej z większych zachodnioniemieckich fabryk samochodów wykazały, że jeden robot zastępuje średnio czterech poprzednio zatrudnionych robotników, powodując jednocześnie konieczność utworzenia jednego nowego stanowiska pracy. Jedną trzecią nowych rodzajów pracy cechowało pogorszenie jakości pracy w porównaniu z zajęciami wykonywanymi poprzednio, co wynikało ze zubożenia treści pracy.

Do pozytywnych aspektów stosowania robotów należy wyeliminowanie szeregu prac prymitywnych i monotonnych. Badania wykazały również, że roboty wpływają na wyeliminowanie ciężkich prac i dają szansę na zmniejszenie zatrudnienia ludzi w niebezpiecznych i wyjątkowo przykrych warunkach, np. przy malowaniu natryskowym. Związki zawodowe w wielu krajach traktują zmiany w produkcji potokowej, spowodowane wprowadzeniem robotów i zautomatyzowanego systemu manipulowania materiałami, jako argument w negocjacjach, w których domagają się zmian w organizacji pracy w kierunku polepszenia tych rodzajów pracy, które pozostają niezmienione nawet w znacznie zautomatyzowanym zakładzie.

Obniżenie kwalifikacji pracowników lub ich przekwalifikowanie

Wraz z likwidacją niektórych prac wykonywanych przez personel niewykwalifikowany, spowodowaną automatyzacją produkcji, występuje również obniżenie poziomu kwalifikacji pracowników zatrudnionych na nowych stanowiskach pracy, powstałych w wyniku zastosowania mikroelektroniki.

Przykład 2. Obrabiarki sterowane numerycznie za pomocą komputera

Obniżanie kwalifikacji prowadzi również do zmian samych umiejętności zawodowych. W przypadku technologii manualnych lub mechanicznych, kwalifikacje są oparte często na doświadczeniu, którego nabycie umożliwia jedynie długa praktyka lub szkolenie. Wraz z pojawieniem się nowej technologii sytuacja ulega zmianie — kwalifikacje opierają się częściej na zdolności do analitycznego i logicznego myślenia, a nie na doświadczeniu.

Przykładowo — obniżenie poziomu kwalifikacji nastąpiło w przemysłach, w których używa się obrabiarek do produkcji małoseryjnej. Pierwsze obrabiarki sterowane numerycznie powstały w latach pięćdziesiątych. Były one sterowane za pomocą znaków dwójkowych, podawanych maszynie na taśmie papierowej lub magnetycznej. Wraz z obniżeniem — dzięki mikroelektronice — kosztu komputerów wprowadzono numeryczne sterowanie komputerowe, w którym minikomputer sterujący jedną lub kilkoma obrabiarkami jest zaprogramowany według wymagań konkretnej pracy, jaka ma zostać wykonana.

W większości krajów uprzemysłowionych operatorzy konwencjonalnych obrabiarek używanych w produkcji małoseryjnej są tradycyjnie bardzo cenieni i znajdują się wysoko w hierarchii prac manualnych. Ich praca wymaga bowiem dużych kwalifikacji i stosunkowo długiego okresu szkolenia. Zadania robotnika obsługującego obrabiarki obejmują udział w planowaniu produkcji danej części, ustawienie maszyny, regulowanie prędkości i działnia obrabiarki zależnie od zmiany takich parametrów, jak np. jakość metalu. Mimo że z pracą tą związane jest porozumiewanie się zarówno z kierownictwem, jak też z kreślarzem i projektantem danego elementu, operator obrabiarki ma sporo swobody i jest w znacznej mierze pracownikiem samodzielnym.

Zastosowanie obrabiarek sterowanych numerycznie za pomocą komputera (ang. computer numericaly controlled, CNC) można uważać za pewną innowację wprowadzoną do procesu produkcyjnego. Na ogół ma ona charakter zastępczy w sensie jej wpływu na umiejętności i kwalifikacje załogi. Analityk i programista rozkładają kwalifikacje operatora obrabiarki na poszczególne elementy logiczne, a komputer zostaje zaprogramowany tak, że z powodzeniem przejmuje całą kontrolę nad funkcjonowaniem obrabiarki, jak czynił to poprzednio robotnik obsługujący maszynę. Sterowanie obrabiarką przechodzi więc z operatora na specjalistę dokonującego analizy pracy wykonywanej przez operatora. Prace operatora zmienia się w pracę polegającą na kontrolowaniu urządzeń sterowanych za pomocą komputera, przy czym z chwilą, gdy systemy CNC staną się bardziej niezawodne, pracę takiego kontrolera będzie można również w znacznej mierze wyeliminować.

Istotnej zmiany należy oczekiwać z chwilą, gdy do obrabiarek sterowanych za pomocą komputera (CNC) dojdzie jeszcze projektowanie wspomagane komputerem (ang. computer aided design — CAD). Zbyteczny stanie się wówczas kontakt między operatorem obsługującym obrabiarkę a projektantem, dotychczas konieczny — zarówno na etapie planowania, jak i projektowania obrabianego elementu. Odpowiednie ustawienie maszyny będzie można przygotować zawczasu, jeszcze w stadium projektowania. Wykona to komputer, który ustawi obrabiarkę zgodnie z wymaganiami technicznymi i specyfiką produkcji danego elementu. Z czasem więc, w zautomatyzowanym warsztacie mechanicznym ludzie nie będą w ogóle potrzebni, komputery, przejmą w całości rolę operatorów obrabiarek.

Zdaniem niektórych specjalistów, proces „odkwalifikowania" przez standaryzację czynności wymagających uprzednio określonych kwalifikacji, przygotowuje niejako pole do kolejnej fazy automatyzacji, w której pewne zawody zostaną całkowicie wyeliminowane. W niektórych krajach prowadzone są obecnie prace badawcze, których ukoronowaniem będzie uruchomienie całkowicie bezzałogowych zakładów mechanicznych.

Wpływ, jaki obrabiarki CNC wywierają na kwalifikacje robotników, zajmujących się obróbką materiałów, jest różny w zależności od różnych zastosowań maszyn sterowanych komputerowo oraz od konkretnych celów produkcyjnych, jakim mają one służyć. Tam, gdzie numeryczne sterowanie za pomocą komputera ma zwiększyć elastyczność procesu i stopień złożoności produktu, wysoki stopień centralizacji planowania i nadzoru może okazać się nieodpowiednią formą organizacji pracy. Można wówczas zastosować inne formy organizacyjne, które będą odpowiadać wprowadzeniu obrabiarek CNC. Dzięki zmniejszeniu, za sprawą rozwoju mikroelektroniki, kosztów komputeryzacji — stosuje się obecnie szeroko w obrabiarkach urządzenia do przetwarzania danych. Oznacza to, że programy robocze dla pojedynczej maszyny można redagować i zmieniać metodą ręcznego wprowadzania danych.

W scentralizowanym systemie obrabiarek CNC, przedstawionym w zarysie na początku niniejszego punktu, redagowaniem programów zajmowałby się programista komputerów z działu przetwarzania danych. W praktyce jednak operatorzy obrabiarek mogą, po przeszkoleniu w zakresie podstawowych umiejętności programisty, sami redagować program do swojej maszyny. A tak naprawdę, w przypadku małoseryjnej albo względnie prostej produkcji, obrabiarki CNC może z powodzeniem programować personel warsztatowy. Wówczas system produkcji stałby się bardziej elastyczny, a produkt końcowy lepszy. Musiałyby jednak wzrosnąć kwalifikacje robotnika obsługującego obrabiarkę, który do posiadanych umiejętności mechanicznych musiałby dodać wiadomości z zakresu informatyki.

Związki zawodowe w niektórych zakładach Norwegii i Szwecji wystąpiły o przyznanie operatorom obrabiarek prawa do samodzielnego programowania maszyn CNC. Spowodowało to konieczność przeszkolenia ich w dziedzinie informatyki, a także wprowadzenia pewnych zmian organizacyjnych, skoro personel warsztatowy miałby teraz bezpośrednio uczestniczyć w produkcji sterowanej przez komputer i mieć dostęp do skomputeryzowanego katalogu schematów części. W jednym z zakładów Norwegii pięciu robotników przeszło roczne szkolenie w zakresie programowania komputerów, uzyskując w ten sposób odpowiednie kwalifikacje do obsługi maszyn CNC.

Brytyjskie związki zawodowe odnotowały przypadki, w których operatorzy obrabiarek pracujący przy obsłudze maszyn CNC nabywali praktycznych umiejętności niezbędnych do programowania komputera — bez specjalnego szkolenia, tylko na podstawie rozmów z inżynierami zatrudnionymi przy instalacji nowych maszyn. Przysposobienie do nowych warunków pracy odbyło się tam bez jakiegokolwiek formalnego szkolenia robotników w dziedzinie informatyki. W końcu jednak załoga zażądała szkolenia oraz przydzielenia dotychczasowym operatorom obrabiarek nowych funkcji związanych z redagowaniem programów dla ich maszyn.

Przykład 3. Drukarstwo

Kolejny przykład innowacji procesu, która prowadzi do obniżenia kwalifikacji personelu, spotykamy w przemyśle poligraficznym. Zjawisko to obserwujemy przy zastępowaniu mechanicznego systemu składu opartego na odlewach z gorącego ołowiu przez skomputeryzowany system oparty na fotoskładzie. Dzięki mikroelektronice koszty zastosowania komputerów obniżyły się do tego stopnia, że ich użycie w tym przemyśle stało się opłacalne. Poważnie zaawansowane są również próby wprowadzenia mikroprocesorów do różnych pomocniczych systemów sterowania w przemyśle poligraficznym, m.in. do automatycznej regulacji nacisku prasy i dopływu farby drukarskiej.

Przemysł gazetowy w wielu krajach europejskich, borykający się z wieloma trudnościami, będący niekiedy na granicy bankructwa, jest kolejnym miejscem zastosowań nowej technologii. Skomputeryzowany system składu umo-

żliwia przygotowanie, adiustację i redakcję oraz ostateczny układ tekstów za pomocą monitora ekranowego (ekranopisu).

W jednym konkretnym przypadku, efektem wprowadzenia fotoskładu sterowanego komputerowo było zmniejszenie liczby zatrudnionych na nocnej zmianie w drukarni gazetowej z 52 do 29 osób. Pracę dwunastu zecerów przejęło tam pięciu operatorów ekranopisów, których zadaniem było przygotowanie tekstów. Dziesięciu korektorów zastąpiono pięcioma operatorami ekranopisów, a pracę dalszych piętnastu osób zajmujących się poprzednio formowaniem kolumn wykonuje teraz siedmiu ludzi.

Zawody zecera, korektora i redaktora technicznego wymagają zasadniczo wysokich kwalifikacji, nabywanych przez lata praktyki, toteż z reguły należą one do lepiej płatnych. Praca tych ludzi, ze względu na swą specyfikę, musi podlegać systematycznej i wymagającej dużego zmysłu organizacyjnego koordynacji. Tych właśnie pracowników można z powodzeniem zastąpić niżej kwalifikowanymi operatorami nowych urządzeń komputerowych. Ich wiedza z zakresu drukarstwa może nie wykraczać poza najbardziej podstawowy poziom. Wyspecjalizowani operatorzy, z kolei, przestaną być w ogóle potrzebni w istniejących już na świecie supernowoczesnych zakładach poligraficznych, gdzie sam dziennikarz wprowadza tekst bezpośrednio do skomputeryzowanego systemu produkującego gotowe gazety. Nowe zawody, wymagające lepszych umiejętności, będą potrzebne w dziedzinie przetwarzania danych.

Coraz większe zagrożenie wynikające z obniżania poziomu kwalifikacji (jak gdyby „odkwalifikowywania") wymaganych od pracowników przemysłu poligraficznego, będące rezultatem zmiany technologii, znajduje wyraz w burzliwych dyskusjach i sporach dotyczących perspektyw drukarzy w wielu krajach. Zimą 1977 w Republice Federalnej Niemiec, na tle przesunięć, które objęły zecerów i drukarzy, powstał poważny problem związany z obniżeniem statusu zawodowego tych ludzi. Spory zakończyły się zagwarantowaniem na osiem lat pracy dla pracowników wykwalifikowanych w zakresie poligrafii oraz — zapewnieniem na sześć lat wynagrodzenia nie niższego niż dotychczas. W Szwecji w podobnej kwestii, osiągnięto porozumienie, na mocy którego pracownicy przemysłu poligraficznego uzyskali gwarancję pełnego zatrudnienia w przypadku przesunięć lub redukcji personalych w ich zakładach.

W Wielkiej Brytanii problem utraty pracy i obniżenia wymagań co do kwalifikacji nasilił się w czasie sporu między brytyjskim związkiem drukarzy a koncernem „Timesa". Efektem sporu było pozbawienie pracy drukarzy tego koncernu od listopada 1978 do października 1979. W Stanach Zjednoczonych dziennik „Washington Post" wprowadził komputerowy system składu bez zgody związków zawodowych. Podczas ciągnących się w następstwie tego faktu sporów, gazetę tworzyło 25 osób kadry kierowniczej obsługujących skomputeryzowany system poligraficzny, zamiast 125 poprzednio zatrudnionych zecerów.

Przykład 4. Praca biurowa

Z rozdziałów Raportu pt. „Technologia stosowana" i „Wpływ na przedsiębiorstwo" wynika jasno, iż należy oczekiwać zastosowania elektronicznego przetwarzania danych w sektorze usług, w zakresie obejmującym także usługi nie poddające się dotychczas automatyzacji. Szczególną uwagę autorzy zwracają na wpływ, jaki mikroelektronika wywiera na pracę biurową. W tej właśnie dziedzinie urządzenia mikroelektroniczne zastąpią prawdopodobnie największą liczbę zatrudnionych. Stanie się to możliwe dzięki obniżeniu kosztów komputeryzacji i wprowadzeniu automatycznego przetwarzania informacji tekstowej, co tradycyjnie odbywało się wyłącznie na zasadzie bezpośredniego udziału człowieka.

Systemy przetwarzania tekstowego są już obecnie kupowane przez liczne instytucje w krajach wysoko uprzemysłowionych i należy się spodziewać, że wpływ tych urządzeń na pracę biurową będzie zwiększał się w ciągu rozpoczętej właśnie dekady.

Wśród zajęć biurowych na czoło wybija się zawód sekretarki. Do jej obowiązków należy głównie korespondencja, obejmująca sporą ilość maszynopisania oraz różne zajęcia administracyjne. Ten rodzaj pracy trudno dokładnie opisać, bowiem zależy on każdorazowo od organizacji pracy z uwzględnieniem np. proporcji czasu przeznaczonego na korespondencję i na zadania administracyjne. Te ostatnie uważa się na ogół za zajęcia wymagające większych umiejętności niż zwykłe przepisywanie na maszynie, które jednak wymaga również znacznych umiejętności w zakresie doboru odpowiedniego układu graficznego i — generalnie — dobrej jakości pracy.

Wprowadzenie systemów przetwarzania tekstowego wpływa w dwojaki sposób na poziom kwalifikacji osób zatrudnionych w typowym sekretariacie. Po pierwsze — może spowodować zmianę organizacji pracy, uwzględniającej rozdział maszynopisania od pracy administracyjnej. Po drugie — narzuca wymaganie zupełnie innych kwalifikacji niż te, którymi dysponuje zawodowa maszynistka.

Przetwarzanie tekstowe wprowadza się często przez powołanie osobnego wydziału, który przejmuje prowadzenie korespondencji całej instytucji. W ten sposób maszyny eksploatuje się intensywniej niż w wypadku utrzymania tradycyjnych struktur poszczególnych sekretariatów, dzięki czemu sprzęt jest wykorzystywany do maksimum. W efekcie zmiany organizacji pracy, osoba obsługująca procesor tekstowy nie musi mieć równie wysokich kwalifikacji jak tradycyjna sekretarka. Nowy zawód polega bowiem zasadniczo na wystukiwaniu tekstu na klawiaturze, nie ma natomiast nic wspólnego z różnorodnymi zadaniami natury administracyjnej. Między osobą piszącą a autorem nie ma żadnego kontaktu, ponieważ teksty do przepisywania odbiera zwykle kierownik wydziału przetwarzania tekstów, on też wyjaśnia również wszystkie ewentualne wątpliwości.

Opisana struktura organizacyjna stanowi jedną z szeregu możliwych. Badania na temat wprowadzenia przetwarzania tekstowego w instytucjach amerykańskich wyodrębniły cztery różne typy struktur. Jedna z nich polega na utworzeniu centralnie administrowanego systemu w ramach danego przedsiębiorstwa, w którym wszystkie sekretarki korespondencyjne przenosi się do ośrodka przetwarzania tekstów, dokąd spływa cała korespondencja do przepisania. Przeciwieństwem tego systemu jest sieć satelitarnych ośrodków, skupiających maszynistki wyszkolone w pisaniu tekstów specjalistycznych (zatrudnia się w nich zresztą zarówno sekretarki administracyjne, jak i korespondencyjne). Trzeci typ struktury polega na utworzeniu rezerwowego centrum, które odciąża tradycyjne sekretarki w razie nadmiaru pracy. Czwartą strukturą jest system zdecentralizowany, w którym urządzenia do przetwarzania tekstów znajdują się w poszczególnych wydziałach. W tym przypadku nie występują ostre podziały specjalizacyjne i sekretarki piszą na maszynach oraz spełniają funkcje administracyjne. Im mniej scentralizowany jest system, tym mniejsza występuje polaryzacja kwalifikacji.

Przetwarzanie tekstowe wymaga od maszynistki nieco innych kwalifikacji. Można pozwolić sobie na mniej staranny układ tekstu, mniejszą dokładność, gorsze wykonanie i mniej uważną korektę, ponieważ wszystko to zrobi maszyna. Do maszynopisania dochodzą jednak dodatkowe umiejętności, które można opanować w ciągu jedno- lub dwutygodniowego kursu szkoleniowego dla operatorów urządzeń do przetwarzania tekstowego. Tam, gdzie w grę wchodzi pisanie tekstów standardowych, można większość czynności zautomatyzować tak, że praca maszynistki będzie wymagać jeszcze mniejszych umiejętności. Przykładowo — pisanie standardowych listów będzie wymagało jedynie umiejętności kodowania.

Badania prowadzone w różnych przedsiębiorstwach ujawniły pewną prawidłowość. Otóż, dobre maszynistki o długiej praktyce zawodowej odczuwają obniżenie poziomu kwalifikacji wymaganych w ich pracy po wprowadzeniu aparatury do przetwarzania tekstów. Natomiast młodsze maszynistki chętnie korzystają z szansy zdobycia nowej umiejętności przy obsłudze bardziej skomplikowanych urządzeń, uważając, że w ten sposób podwyższają swoje poprzednie kwalifikacje. Przedstawiciele jednego z brytyjskich związków zawodowych urzędników stwierdzili jednak, że z chwilą, gdy pracownicy przyzwyczają się do procesorów tekstowych i zniknie pierwotny zachwyt nowoczesnością, ich operatorzy rozczarują się i zaczną odczuwać mankamenty większej specjalizacji zawodowej.

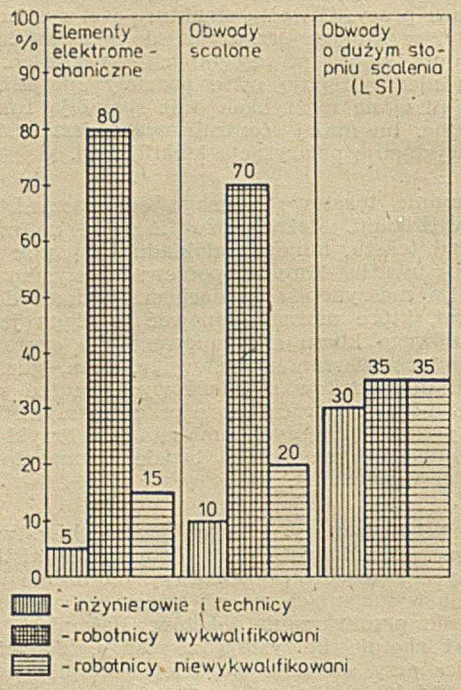
Przy wprowadzaniu nowej technologii do urzędu można wybierać model organizacji pracy. O wyborze takiego modelu powinny decydować względy społeczne, nie zaś techniczne — od nich bowiem zależy choćby rozstrzygnięcie kwestii, czy należy utworzyć bardzo scentralizowany wydział przetwarzania tekstowego, odpowiedzialny wyłącznie za prowadzenie korespondencji, a innym wydziałom pozostawić ich własne sekretarki, zajmujące się już tylko sprawami administracyjnymi. Równie dobrze bowiem można przyjąć taki model organizacyjny, w którym nowo zainstalowane procesory tekstowe oddaje się do dyspozycji wszystkich sekretariatów — po to, by odciążały w jednakowym stopniu wszystkie sekretarki, pozostawiając im więcej czasu na funkcje administracyjne. Taki model nieźle sprawdza się przy zastosowaniu zasady rotacji pracowników.

W przyszłości, gdy procesory tekstowe będą znacznie tańsze, utracą znaczenie względy natury ekonomicznej, narzucające maksymalne wykorzystanie tych urządzeń.

Przykład 5. Mikroelektronika zamiast mechaniki

Przykłady „odkwalifikowania" mogą dotyczyć również wprowadzenia mikroelektroniki zamiast mechaniki. Zastosowanie nowej technologii rzutuje poważnie na zatrudnienie i kwalifikacje zawodowe pracowników, m.in. w tych przemysłach, w których elementy mechaniczne i elektromechaniczne zostały zastąpione elementami mikroelektronicznymi. Zmniejszenie liczby części składowych dzięki mikroelektronice prowadzi do uproszczenia procesu montażu i zmniejszenia wkładu pracy potrzebnej do wyprodukowania określonych artykułów, co scharakteryzowano dokładniej w rozdziale pt. „Wpływ na przedsiębiorstwo". Do urządzeń, które dzięki mikroelektronice uległy największej zmianie, należą m.in. aparaty telegraficzne, maszyny do szycia, zegarki elektroniczne, kalkulatory, kasy sklepowe itd.

W zegarmistrzostwie, na wyprodukowanie tradycyjnego zegarka składa się ponad tysiąc oddzielnych czynności wymagających operacji mechanicznych i ręcznych. Wszystkie one zostały zastąpione przez montaż pięciu podstawowych elementów wchodzących w skład zegarka elektronicznego, a mianowicie: baterii, kryształu kwarcu, obwodu scalonego, wskaźnika i koperty. W rezultacie zawód dawnego zegarmistrza może niebawem okazać się zbyteczny.



Zmiany składu załogi zakładu produkcji elementów elektronicznych, w wyniku wprowadzenia zmian technologicznych. Źródło: Haroldo Correa de Mattos, Technology and Developing Countries in the ITU (Technologia a kraje rozwijające się w Międzynarodowej Unii Telekomunikacji), III Światowe Forum Telekomunikacji, Genewa 1979, tom I, s. 11

Innego przykładu dostarcza przemysł produkujący aparaty telegraficzne. W jednym z rozwiązań aparatu elektronicznego, za pomocą mikroelektroniki zastąpiono 936 części wchodzących w skład modelu elektromechanicznego. Wskutek tej zmiany zapotrzebowanie na wyszkolonych pracowników zmalało z 82 do 35% (w stosunku do całej załogi fabryki). Jednocześnie wysoko wykwalifikowany personel techniczny zwiększył się z 2 do 30%. W Republice Federalnej Niemiec zaobserwowano szereg przykładów „odkwalifikowania" w sektorze budowy mszyn. W niektórych przypadkach robotnikom pracującym na linii montażowej obniżono zaszeregowanie o cztery kategorie w systemie obejmującym 12 kategorii. W innych przypadkach rady zakładowe zdołały ograniczyć obniżenie kategorii tylko o dwie, a w jednym — załoga podjęła akcję strajkową w ramach protestu, co skończyło się lokautem, wskutek którego ucierpiało kilkaset tysięcy robotników.

W dziedzinie remontów, napraw i konserwacji następuje również spadek kwalifikacji załóg — wszędzie tam, gdzie systemy mikroelektroniczne wprowadza się na miejsce eletromechanicznych. Sprzęt elektroniczny stawia mniejsze wymagania konserwacyjne niż sprzęt elektromechaniczny, bowiem mniej jest w nim podzespołów i mniej ruchomych elementów. Mikroelektronika stwarza również możliwość zastosowania urządzeń samowykrywających uszkodzenia. Prowadzi to do „odkwalifikowania" elektryka z ekipy remontowej, który nie musi mieć tak wysokich kwalifikacji jak poprzednio. Konserwacja polega teraz na dość łatwym zabiegu wymiany uszkodzonych znormalizowanych podzespołów (modułów). Jednocześnie, jak wyjaśniono, na zamieszczonym rysunku, konserwacja sprzętu mikroelektronicznego wymaga pewnej liczby wysoko wykwalifikowanych specjalistów w dziedzinie elektroniki.

Przykład 6. Produkcja półprzewodników

Międzynarodowe badania przeprowadzone w firmach elektronicznych na zlecenie Organizacji Współpracy i Rozwoju Gospodarczego (OECD) w 1977 roku wykazały, że w zakładach produkujących półprzewodniki wymagania dotyczące kwalifikacji pracowników zmiejszały się gwałtownie, a produkcja układów o dużym stopniu scalenia (LSI) zaczynała coraz bardziej przypominać przebieg procesu chemicznego. W raporcie sporządzonym na podstawie tych badań stwierdzono, że generalnie dochodzi do polaryzacji kwalifikacji zawodowych. Z jednej strony — istnieje wysoko wykwalifikowany personel wyszkolony w projektowaniu, testowaniu i kontroli technicznej, a z drugiej — grupa niewykwalifikowanych operatorów maszyn, natomiast w ogóle zanika kategoria średnio wykwalifikowanych robotników produkcyjnych. Z badań wynika również, że polaryzacja poziomu kwalifikacji jest związana z eksportem niewykwalifikowanej siły roboczej do innych krajów, gdzie można znaleźć zatrudnienie przy montażu taśmowym. Zatem, znaczenie rozwoju technologicznego dla zatrudnienia w skali kraju przejawia się w stworzeniu niewielkiej liczby miejsc pracy wymagających wysokich kwalifikacji. Na rysunku przedstawiono zmiany składu załogi na wydziale produkcji elementów elektronicznych, które nastąpiły wskutek wprowadzenia układów o dużym stopniu scalenia (LSI).

Podwyższenie kwalifikacji

Stwierdziliśmy już, że zastosowanie mikroelektroniki może prowadzić do koncentracji wymaganych kwalifikacji w mniejszej liczbie nowych zawodów, tych które są związane przede wszystkim z posługiwaniem się komputerem. W związku z postępem mikroelektroniki zwiększa się zapotrzebowanie na te właśnie zawody. Można wśród nich wyodrębnić z grubsza trzy kategorie. Pierwsza obejmuje głównie inżynierów elektroników oraz specjalistów w dziedzinie sytemów sterowania i produkcji oraz układów dynamicznych. Drugą grupę tworzą specjaliści w dziedzinie operacji logicznych, systemów i oprogramowania. Do trzeciej zaliczamy personel mający ogólną orientację w dziedzinie przetwarzania danych.

W początkowym okresie rozwoju komputeryzacji, kiedy do użycia wchodziły duże sytemy komputerowe, stworzono dość pokaźną liczbę specjalistycznych zawodów opartych na powtarzalnych czynnościach w zakresie przygotowywania i wprowadzania danych. Usprawnienie procesów

wprowadzania i wyprowadzania informacji, dzięki zastosowaniu klawiatury i monitora ekranowego, umożliwiło zrezygnowanie ze znacznej części obsługi, ponieważ bezpośrednim wykonywaniem tych czynności zajmują się bezpośredni użytkownicy komputera. Pracownicy zajmujący się jedynie przygotowywaniem danych należeli na ogół do stosunkowo niskiej kategorii zawodowej.

Powstawanie miejsc pracy dla przedstawicieli nowych zawodów nie musi wcale iść w parze z eliminacją niektórych specjalności lub obniżaniem wymagań kwalifikacyjnych w tym samym przedsiębiorstwie lub gałęzi gospodarki. Wykwalifikowany personel komputerowy znajduje zatrudnienie w trzech głównych obszarach: w przedsiębiorstwach wykorzystujących komputery lub mikroelektronikę w procesach produkcyjnych, w przedsiębiorstwach produkujących komputery i sprzęt mikroelektroniczny oraz w bardzo szybko rozwijających się firmach oferujących usługi konsultacyjne i konserwacyjne w zakresie sprzętu i oprogramowania oraz usługi przetwarzania w trybie abonenckim.

Badania przeprowadzane w Japonii, do których nawiązano w rozdziale pt. „Wpływ na przedsiębiorstwo", dotyczyły zmian w zatrudnieniu zarówno w przedsiębiorstwach produkujących, jak i wykorzystujących urządzenia mikroelektroniczne. Wyniki badań dowodzą, że nowe stanowiska pracy dla wysoko wykwalifikowanych specjalistów w zakresie rozwoju i projektowania powstają w zakładach produkujących automatyczne urządzenia i sprzęt mikroelektroniczny powszechnego użytku. Natomiast w przedsiębiorstwach, które korzystają z takich urządzeń, zatrudnienie zmniejsza się zarówno w przypadku pracowników wykwalifikowanych, jak i niewykwalifikowanych. Te same badania stanowią podstawę przewidywań, że w latach 1972—1985 zapotrzebowanie na inżynierów wyspecjalizowanych w programowaniu będzie zwiększać się średnio o 12% rocznie.

Z cytowanych już badań przeprowadzonych w firmach zachodnioniemieckich wynika, że wprowadzenie robotów przemysłowych spowodowało zwiększenie liczby personelu wykwalifikowanego w danym sektorze, przy jednoczesnym zmniejszeniu liczby robotników niewykwalifikowanych. Podobną sytuację stwierdzono w przemyśle drukarskim. Wprowadzenie skomputeryzowanego składu wpłynęło na zwiększenie personelu wyspecjalizowanego w przetwarzaniu danych, powodując przemieszczanie wykwalifikowanych składaczy i zecerów.

Równolegle występujące zjawiska zmniejszania się zapotrzebowania na istniejące zawody oraz zwiększenia zapotrzebowania na nowe specjalności można również zaobserwować w dziedzinie konserwacji maszyn. Wykazaliśmy już, że w miarę jak mikroelektronika wypiera bardziej przestarzałe urządzenia elektromechaniczne, w ekipach remontowych potrzeba coraz mniej elektryków. Chociaż zasady konserwacji sprzętu elektronicznego, są prostsze niż w przypadku sprzętu elektromechanicznego, to jednak ciągle będzie zwiększać się zapotrzebowanie na konserwatorów wyspecjalizowanych w obsłudze urządzeń elektronicznych, zwłaszcza mających wiedzę z zakresu pneumatyki i hydrauliki, w zależności od charakteru zautomatyzowanego zakładu.

Rozstrzygnięcie kwestii czy wraz z wprowadzeniem elektroniki powstają warunki do podwyższenia czy do obniżenia kwalifikacji załogi, zależy częściowo od możliwości znalezienia dla powstałych nadwyżek siły roboczej stanowiska pracy, na których przesunięty personel stykałby się z elektroniką i obsługą komputerów. Naturalnie, poważniejsze problemy występują wówczas, gdy zmianą kwalifikacji trzeba objąć robotników starszych wiekiem. W pewnych okolicznościach najlepszym rozwiązaniem zmniejszającym społeczne koszty, które wskutek zastosowania nowej technologii ponoszą starsi robotnicy, może być wprowadzenie wcześniejszych emerytur na poziomie poprzednich dochodów.

Pomiędzy różnymi grupami zawodowymi mogą też istnieć pewne strukturalne bariery utrudniające ewentualne przekwalifikowanie pracowników. Przyczyną bezruchu pomiędzy różnymi grupami zawodowymi może być fakt, że redukcją są objęci wykwalifikowani robotnicy fizyczni, natomiast nowe miejsca pracy tworzy się dla inżynieryjno-technicznych pracowników umysłowych. Istotne mogą być też różnice wynikające ze zmiany technologii, stwierdzono bowiem, że inżynierowie mechanicy nie zawsze są dobrymi inżynierami elektronikami.

Jak już wspomniano, wprowadzenie nowej technologii opartej na mikroelektronice szczególnie odczuwają starsi wiekiem robotnicy. Mają oni trudności z przystosowaniem się do nowych warunków pracy, wymagających przygotowania elektronicznego, gdyż dotąd wykonywali zawód oparty na rozwiązaniach typu mechanicznego. Mimo że stosuje się różne sposoby łagodzenia wpływu zmiany technologii na profil kwalifikacji załogi, to na ogół istnieje poważne zagrożenie trwałej utraty pracy przez znaczną liczbę robotników starszych wiekiem. Należałoby przeto zapewnić im możliwość wcześniejszego odejścia na emeryturę na zasadzie dobrowolności oraz zagwarantować odpowiednio wysokie stawki emerytalne.

Gwałtowny wzrost zapotrzebowania na specjalności związane z obsługą komputerów doprowadził do generalnego niedoboru informatyków, mimo wyraźnie zwiększonego ogólnego poziomu bezrobocia. Oficjalne badania finansowane przez rząd brytyjski wykazały, że w 1980 roku w tym kraju było zatrudnionych 275 tys. osób z wykształceniem informatycznym, natomiast w różnych sektorach gospodarki brakowało ogółem 25 tys. informatyków. W przemyśle produkującym komputery brakowało 16 tys. programistów i analityków, co stanowi 15% wszystkich zatrudnionych w tej gałęzi przemysłu. Notowano również stuprocentowy niedobór inżynierów elektroników. Z danych szacunkowych wynika, że podobną sytuację obserwujemy na rynku pracy dla kadry komputerowej we Francji i RFN.

Tłumaczyła EWA WOYDYŁŁO

Finansowanie prenumeraty

Wyjaśnienie Ministra Finansów w sprawie pokrywania ze środków funduszu postępu technicznego i organizacyjnego kosztów zakupu czasopism technicznych na potrzeby ośrodków informacji naukowo-technicznej i ekonomicznej oraz bibliotek zakładowych (p. II okł.) zostało oparte na interpretacji następujących przepisów:

- § 15 ust. 1 pkt 1 rozporządzenia Rady Ministrów z dnia 28.6.80 w sprawie szczegółowych zasad gospodarki finansowej przedsiębiorstw państwowych oraz dostosowania przepisów ustawy o gospodarce finansowej przedsiębiorstw państwowych do specyfiki niektórych przedsiębiorstw (Dz. U. nr 22, poz. 156).
- § 2 uchwały nr 282 Rady Ministrów z dnia 30.12.82 w sprawie zasad finansowania prac badawczych oraz zasad działalności jednostek badawczych w 1983 r. (M. P. nr 4 z 1983 r., poz. 26).

Kolejnym językiem, który przedstawiamy na łamach INFORMATYKI, jest BASIC. Taka decyzja może wydawać się kontrowersyjna — wszak nie jest to język nowoczesny, który charakteryzuje się czymś wyjątkowym. Wiemy jednak, że obecnie jest to podstawowy język wysokiego poziomu do programowania mikrokomputerów — w zasadzie każdy użytkownik mikrokomputera musi go znać. Tylko — czy rozpowszechnienie języka jest wystarczającym powodem, aby poświęcać mu aż trzy artykuły planowane w niniejszym cyklu?

Podstawowa przyczyna jest inna. Na świecie dostrzeżono rolę BASICA w programowaniu komputerów i podjęto prace nad jego standaryzacją. Jak duże znaczenie dla efektywnego programowania ma istnienie powszechnie uznanej normy języka, nie trzeba oddzielnie uzasadniać — przede wszystkim chodzi o zapewnienie daleko posuniętej wymienności programów. Sądzimy, że samo przedstawienie języka i nawiązanie do prac organizacji normalizacyjnych będzie bardzo pożyteczne dla Czytelników i użytkowników BASICA w Polsce. Wszak obecne zróżnicowanie wersji i implementacji języka jest bardzo niekorzystne.

Dodatkowy, ważny aspekt tej serii artykułów, to ich aktualność. Informacje przedstawione w dwóch następnych częściach, dotyczące tzw. modułu rdzeniowego i rozszerzeń standardowych, są oparte na nie opublikowanym jeszcze Raporcie. Ich treść obejmuje znacznie szerszy obszar tematyczny niż dwie wydane dotąd w Polsce książki o BASICU (cytowana w artykule książka jego autorów i wydana pod tym samym tytułem przez WNT książka Z. Czecha, K. Nałęckiego i S. Wołka). Szczególnie interesujące są próby rozszerzenia standardu BASICA w stronę programowania systemów czasu rzeczywistego i systemów graficznych.

WACŁAW ISZKOWSKI
MAREK MANIECKI

Instytut Informatyki
Politechnika Warszawska

Całość, składająca się z trzech odcinków, będzie zatem w istotny sposób różnić się od tego, co do tej pory wiadomo w Polsce o BASICU. Dlatego też sądzimy, że zainteresuje szersze grono Cztelników. (Red.)

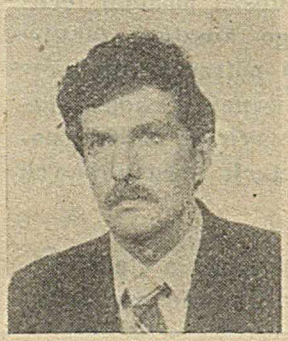
Standaryzacja języka BASIC (1)

Norma języka minimalnego

Język BASIC (Beginners All-purpose Symbolic Instruction Code) został zaprojektowany w 1964 roku przez profesorów J. G. Kemeny'ego i T. E. Kurtza z Dartmouth College w USA. Początkowo był przeznaczony głównie dla programistów nieprofesjonalnych. Dzięki swej prostocie, uniwersalności, a także możliwości interakcyjnego wykorzystania, BASIC szybko stał się bardzo popularny. Brak normy języka oraz stale rozszerzający się zakres zastosowań spowodowały powstawanie nowych jego wersji oraz translatorów [4]. Niestety, rozszerzenia — chociaż niekiedy bardzo interesujące — często były niezgodne z przyjętymi podstawami konstrukcji języka oraz uniemożliwiały przenoszenie programów.

Dopiero w roku 1974, pod wpływem fali nowych zastosowań BASICA jako podstawowego języka dla mikrokomputerów oraz komputerów osobistych, rozpoczęto prace nad jego standaryzacją. Działaniom tym patronuje organizacja ANSI (American National Standards Institute) oraz ECMA (European Computer Manufacturers Association). W pierw-

Dr inż. WACŁAW ISZKOWSKI ukończył w 1972 r. studia na Wydziale Elektroniki Politechniki Warszawskiej, a w 1977 r. uzyskał stopień doktora nauk technicznych. Od 1972 r. pracuje w Instytucie Informatyki Politechniki Warszawskiej, prowadząc zajęcia dydaktyczne i prace naukowo-badawcze związane z językami programowania, systemami operacyjnymi i programowaniem współbieżnym.



Dr inż. MAREK MANIECKI ukończył w 1972 r. studia na Wydziale Elektroniki Politechniki Warszawskiej, a w 1977 r. uzyskał stopień doktora nauk technicznych. Od 1972 r. pracuje w Instytucie Informatyki Politechniki Warszawskiej, prowadząc zajęcia dydaktyczne i prace naukowo-badawcze związane z językami programowania, systemami operacyjnymi i programowaniem współbieżnym.

szym okresie, po prawie czteroletnich dyskusjach amerykańskiego komitetu X3J2 oraz europejskiego TC 21, w 1977 roku przygotowano normę minimalnego języka BASIC (Minimal BASIC) [1, 6], która w roku 1980 została zaakceptowana przez ISO (International Organisation for Standardization) [3].

Obecnie trwają prace nad normą pełnego języka BASIC. W definicji języka przyjęto rozszerzoną wersję minimalną jako moduł rdzeniowy (ang. core module), który może być następnie uzupełniany o dodatkowe moduły: zarządzania plikami, grafiki komputerowej, systemu czasu rzeczywistego, obliczeń stałoprzecinkowych oraz moduł redagowania. Według aktualnej propozycji normy języka [2, 5], wszystkie wymienione moduły tworzą język BASIC. Zakończenie prac nad standaryzacją jest przewidziane na koniec bieżącego roku.

Podstawowym założeniem propozycji normalizacyjnej jest wymaganie zgodności programu z regułami syntaktycznymi podanymi w normie. Wymaga się też zgodności implementacji języka z normą, polegającej na akceptacji programów zgodnych z normą, odrzuceniu niezgodnych oraz na uzupełnieniu normy o elementy określone w niej jako zależne od implementacji.

W pierwszej części artykułu przedstawiamy elementy normy języka minimalnego. Ze względu na konieczność skrótowego opisu założyliśmy, że większość Czytelników zna jedną z wersji języka BASIC i skoncentrowaliśmy się na krótkim omówieniu elementów języka zaliczonych do wersji minimalnej, bez podawania pełnego opisu składni i semantyki.

TEKST PROGRAMU

Program w minimalnym języku BASIC składa się z ponumerowanych linii o maksymalnej długości 72 znaków. Numery linii są liczbami naturalnymi nie większymi od 9999. Każda linia zawiera pojedynczą instrukcję języka oznaczoną przez słowo kluczowe znajdujące się bezpośrednio po numerze linii.

Kolejność wprowadzania linii programu może być dowolna, a o kolejności wykonywania instrukcji decyduje wartość numeru linii. Linie o tym samym numerze mogą być zdefiniowane wielokrotnie powodując skasowanie poprzednich definicji. Ostatnią instrukcją programu musi być instrukcja końca (o słowie kluczowym END), oznaczająca koniec treści programu.

Dopuszcza się używanie jedynie dużych liter. Oprócz numeru linii i słów kluczowych instrukcji elementami linii są identyfikatory zmiennych, stałe numeryczne i napisowe, identyfikatory funkcji oraz symbole operatorów. Separatorem elementów w linii jest spacja.

OPERACJE NUMERYCZNE

Dziedziną operacji numerycznych są z założenia liczby typu zmiennoprzecinkowego. W implementacji języka należy zapewnić zakres obliczeń numerycznych o wartości bezwzględnej z przedziału 1E-38..1E38. Wartość bezwzględna mniejsza od 1E-38, ze względu na dokładność obliczeń na liczbach typu zmiennoprzecinkowego, jest równa zeru.

Stałe numeryczne mogą być zapisywane jako liczby całkowite bądź zmiennoprzecinkowe, w postaci ułamkowej lub wykładniczej. Wymagana jest możliwość zapisu liczb z precyzją co najmniej sześciu cyfr znaczących.

Zmienne numeryczne proste są identyfikowane pojedynczą literą lub literą z cyfrą. Zmiennych tych nie trzeba deklarować. Podane identyfikatory mogą też oznaczać zmienne numeryczne tablicowe jedno- lub dwuwymiarowe. Do deklaracji tablicy służy instrukcja deklaracji (DIM), w której po identyfikatorze podaje się liczbę elementów dla każdego z rozmiarów. Bazą numeracji elementów jest indeks równy 1 lub — po zastosowaniu instrukcji OPTION BASE 0 — indeks równy 0 (problem sposobu specyfikacji bazy dla tablic dyskutowano na forum wspomnianych komitetów przez trzy lata!).

Linia:

```
10 DIM A(12), B(5, 6)
```

deklaruje wektor A o 12 elementach oraz macierz B o 5 wierszach i 6 kolumnach.

Instrukcja deklaracji (o ile jest użyta) musi poprzedzać w programie pierwsze użycie zmiennej tablicowej. Brak takiej deklaracji powoduje przyjęcie deklaracji standardowej tablicy jednowymiarowej 10-elementowej lub dwuwymiarowej 100-elementowej (lub odpowiednio 11- i 121--elementowej).

Identyfikator elementu zmiennej numerycznej tablicowej składa się z identyfikatora tablicy oraz z jednego lub dwóch wyrazów (w nawiasach okrągłych), których część całkowita określa indeksy elementu.

Argumentami wyrażeń numerycznych są stałe, zmienne oraz wywołania funkcji numerycznych. Obliczenie wartości wyrażenia jest zgodne z priorytetem operatorów, kolejnością od lewej do prawej oraz z uwzględnieniem podwyrażeń ujętych w nawiasy okrągłe.

Instrukcja podstawienia jest zawsze identyfikowana słowem kluczowym LET. Instrukcja:

```
30 LET X = (5*E-1)*(5E-1+Z)
```

przypisuje zmiennej X wartość wyrażenia (5*E-1)*(.5+Z), w którym 5E-1 jest zapisem wykładniczym stałej.

FUNKCJE NUMERYCZNE

W języku zdefiniowano 10 standardowych funkcji numerycznych (trygonometrycznych, logarytmicznych, arytmetycznych) o trzyliterowych identyfikatorach i jednym parametrze. Jedenasta funkcja (RND) jest bezparametrowa, a jej wartością jest kolejna liczba pseudolosowa pobrana z ciągu liczb pseudolosowych, stałego dla danej implementacji. Z działaniem tej funkcji wiąże się instrukcja RANDOMIZE przesuwająca pseudolosowo początek ciągu liczb pseudolosowych.

Oprócz funkcji standardowych istnieje możliwość zdefiniowania maksymalnie 26 własnych funkcji w programie. Definicja funkcji (DEF) wiąże pojedynczą literę ze standardowym przedrostkiem, określa identyfikator parametru i postać wyrażenia numerycznego. Parametr (pojedynczy) funkcji może być oznaczony dowolnym dopuszczalnym identyfikatorem zmiennej numerycznej, przy czym oznacza on zawsze inny obiekt niż zmienna o tym samym identyfikatorze. Wartość funkcji jest określana przez wyrażenie, w którym mogą występować dowolne zmienne numeryczne i zdefiniowane wcześniej funkcje. Definicja funkcji musi zmieścić się w jednej linii o numerze mniejszym niż numer linii, w której występuje jej wywołanie. Wywołanie funkcji standardowej lub definiowanej jest dopuszczalne w wyrażeniu przez podanie jej identyfikatora i wartości argumentu (w nawiasach okrągłych). Przykładowa definicja funkcji może mieć postać:

```
50 DEF FNX(X) = (X*X+1)/(X*X-1)+Y
```

gdzie X jest parametrem formalnym, a Y — prostą zmienną numeryczną. Wywołanie funkcji może nastąpić w instrukcji podstawienia:

```
100 LET Z = COS(FNX(2.5)+FNX(Y))
```

gdzie COS jest identyfikatorem funkcji standardowej.

INSTRUKCJE STERUJĄCE

Instrukcjami sterującymi są: instrukcja skoku (identyfikowana słowem kluczowym GOTO lub GO TO), warunku (IF..THEN), przełączania (ON..GOTO), stopu (STOP) oraz instrukcja skoku do podprogramu (GOSUB lub GO SUB) i powrotu (RETURN). Instrukcje te zmieniają naturalną sekwencję wykonywania linii programu wskazując kolejną linię do wykonania. Numer linii wymieniony jako argument musi istnieć w programie (może być numerem linii komentarza).

Instrukcja warunku:

IF **wyrażenie_relacyjne** THEN **numer_linii**

dla relacji spełnionej dokonuje przejścia do wskazanej linii. W przeciwnym przypadku wykonywana jest instrukcja z linii następnej. Wyrażenie relacyjne występujące po słowie kluczowym IF jest relacją wartości dwóch wyrażeń numerycznych (dla relacji równości, także — dwóch wyrażeń napisowych).

Poniższy fragment programu oblicza wartość przykładowej funkcji nieciągłej.

```
60 IF X>1 THEN 90
70 LET Y=X
80 GO TO 100
90 LET Y=LOG(X)
100 REM NASTEPNE INSTRUKCJE
```

Instrukcja przełączania:

ON **wyrażenie_numeryczne** GO TO **lista_numerów_linii**

dokonuje przejścia do linii o numerze podanym na liście, równym części całkowitej wartości wyrażenia. Brak pozycji na liście dla danej wartości powoduje zakończenie wykonywania programu. Załóżmy przykładowo, że dla każdej wartości całkowitej I z przedziału -4..2 należy wykonać inny fragment programu. Można wtedy użyć następującej instrukcji przełączania:

```
120 ON I+5 GO TO 400, 500, 600, 700, 800, 900, 1000
```

Instrukcja STOP kończy wykonywanie programu w dowolnym miejscu bezwarunkowo, w odróżnieniu od instrukcji END kończącej program po wykonaniu instrukcji w ostatniej linii (o największym numerze).

Instrukcja:

GOSUB **numer_linii**

dokonuje przejścia (z zapamiętaniem śladu) do wskazanej linii określającej początek podprogramu. Nie ma jednak żadnej możliwości przeniesienia wartości argumentów, a zmienne występujące w liniach objętych działaniem podprogramu są częścią całego zbioru zmiennych programu. Instrukcja RETURN dokonuje powrotu do linii następującej po instrukcji GOSUB. Dopuszcza się wielokrotnie, rekurencyjne wykonywanie instrukcji GOSUB, przy zapewnieniu odpowiedniej liczby instrukcji RETURN.

Poniższy podprogram dokonuje rekurencyjnego obliczania N-tego elementu ciągu Fibonacciego.

```
1000 LET N=N-1
1010 IF N<2 THEN 1070
1020   GOSUB 1000
1030   LET X9=F+F1
1040   LET F1=F
1050   LET F=X9
1060   RETURN
1070 LET F1=1
1080 LET F=1
1090 RETURN
```

Wywoływanie podprogramu w celu obliczenia 30 elementu ciągu ma postać:

```
500 LET N=30
510 GOSUB 1000
520 ...
```

Wynik działnia podprogramu jest przypisany zmiennej F.

INSTRUKCJA ITERACYJNA

Instrukcja iteracyjna cyklu wyliczanego FOR...NEXT obejmuje instrukcje występujące po instrukcji wyliczania: FOR **zmienna_wyliczana** = **wyrażenie** TO **wyrażenie** STEP **wyrażenie** do odpowiadającej jej instrukcji następnika:

NEXT **zmienna_wyliczana**

Elementem wiążącym wymienione instrukcje jest identyfikator zmiennej wyliczanej.
Działanie przykładowej instrukcji cyklu wyliczanego

```
100 FOR I=1 TO N STEP K
...
170 NEXT I
```

można zapisać z wykorzystaniem instrukcji podstawienia i warunku:

```
100 LET granica=N
101 LET krok=K
102 LET I=1
103 IF (I—granica)*SGN(krok)>0 THEN 172
...
170 LET I=I+krok
171 GOTO 103
172 REM..KONIEC CYKLU
```

W tym przypadku zmienne **granica** i **krok** są zmiennymi lokalnymi ustalanymi zależnie od implementacji (np. w rejestrach roboczych). Instrukcja cyklu wyliczanego FOR..NEXT może być zawarta całkowicie w instrukcji FOR..NEXT wykorzystującej inną zmienną wyliczaną. Nie wolno wykonywać instrukcji objętych cyklem nie przechodząc uprzednio przez instrukcję FOR rozpoczynającą ten cykl.

OPERACJE NAPISOWE

Dziedziną operacji napisowych jest sekwencja znaków kodu ASCII o długości od 0 do 18. Stała napisowa jest ciągiem znaków ujętym w cudzysłów. Zmienne napisowe są identyfikowane literą uzupełnioną znakiem dolara $, tzn. A$ i A oznaczają różne zmienne. Zmiennych napisowych również nie trzeba deklarować. W języku minimalnym nie ma tablicowych zmiennych napisowych.

Wyrażeniem napisowym jest pojedyncza stała lub zmienna napisowa. Instrukcja LET umożliwia podstawianie napisów jako wartości zmiennych. Przy użyciu wyrażeń relacyjnych (instrukcja IF) można porównywać dwa wyrażenia napisowe, tj. sprawdzać, czy są identyczne co do długości i kolejnych znaków.

W poniższej sekwencji linii:

```
600 LET A$="ALFA"
610 ...
...
800 IF A$="ALFA" THEN 830
810 REM TO NIE JEST ALFA
820 GOTO 840
830 REM A TO JEST ALFA
840 ...
```

następuje sprawdzenie (linie 800—830), czy zmienna napisowa A$ ma wartość „ALFA".

INSTRUKCJE WPROWADZANIA I WYPROWADZANIA

Do wprowadzania wartości danych do programu służą instrukcje wczytywania (INPUT i READ), deklarowania wartości (DATA) i organizacji danych (RESTORE).

Instrukcja:

INPUT **lista_zmiennych**

wprowadza z terminala wartość podstawianą pod kolejne zmienne wymienione w linii. Można wprowadzać zarówno wartości numeryczne, jak i napisowe — przy spełnieniu warunku zgodności typu wartości z typem aktualnie podstawianej zmiennej. Wykonaniu tej instrukcji może towarzyszyć wyprowadzenie na terminal znaku zachęty (ang. prompt) mającego postać ustaloną w implementacji języka. Przykładowo, wprowadzenia danych można dokonać przy użyciu instrukcji:

```
100 INPUT I$, N$, D, M, R, M$
```

gdzie kolejne zmienne oznaczają: imię, nazwisko, dzień, miesiąc, rok i miejsce urodzenia.

Instrukcja DATA umieszcza w treści programu sekwencję danych, które mogą być stałymi numerycznymi lub napisowymi. Sekwencje tworzy się ze wszystkich instrukcji DATA występujących w programie, w kolejności numerów linii. Stałe napisowe, występujące na liście instrukcji DATA, muszą być ujęte w cudzysłowy, jeżeli są poprzedzone lub uzupełnione spacjami.

Do nadania zmiennym programu wartości z sekwencji danych służy instrukcja READ, działająca analogicznie jak INPUT. Przed rozpoczęciem wykonywania programu oraz po każdym wykonaniu instrukcji RESTORE, wskaźnik pobierania danych z sekwencji jest ustawiany na jej początek.

Ilustracją działania opisanych instrukcji jest poniższy fragment programu rozpoznający słowa kluczowe języka BASIC.

```
100  INPUT  S$
110  RESTORE
120  FOR I = 1 TO 18
130    READ  W$
140    IF  W$ = S$  THEN  200
150  NEXT I
160  REM SŁOWO NIEROZPOZNANE
...
200  REM ROZPOZNANO I-TE SŁOWO
...
1000 DATA DATA, DEF, DIM, GOSUB, END
1010 DATA GOTO, IF, INPUT, LET, ON, OPTION
1020 DATA PRINT, RANDOMIZE, READ, REM
1030 DATA RESTORE
1040 DATA RETURN, STOP
```

Wyprowadzenia wyników działania programu dokonuje instrukcja wydruku:

PRINT **lista_wyprowadzania**

przy czym na liście wyprowadzania znajdują się wyrażenia numeryczne lub napisowe oddzielone separatorami (średnikiem lub przecinkiem). Postać wyprowadzanych wartości wyrażeń numerycznych jest dobierana w zależności od wartości liczby. Średniki i przecinki występujące na liście wyprowadzania instrukcji dokonują łączenia (sklejania) wyprowadzanych napisów (średnik) lub tabulacji w kolumnach co 15 znaków (przecinek). Brak separatora na końcu listy powoduje wyprowadzenie znaku końca linii (ustalonego w danej implementacji). Elementem listy wyprowadzania może być też wywołanie funkcji TAB dokonującej tabulacji zgodnie z podanym dla niej argumentem. Przykładowy fragment programu:

```
 90  PRINT "PODAJ WARTOŚĆ"
100  INPUT X
...
390  LET  X$ = "X"
400  PRINT X$, „=" "LOG("XS,") = ",LOG(X)
```

powoduje dla X = 1 wyprowadzenie napisu
X = 1 LOG(X) = 0

* * *

Powyższy opis jest jedynie przypomnieniem podstawowych właściwości języka BASIC w wersji minimalnej. W drugiej części artykułu przedstawimy bardziej szczegółowo propozycje nowej, pełnej normy języka — wspomnianego już modułu rdzeniowego.

LITERATURA

[1] American National Standard for Minimal BASIC. Report ANSI X3.60, American National Standards Institute, New York, 1978

[2] Draft American National Standard for BASIC. Report ANSI X3J2/82—17. Americah National Standards Institute, New York, October 1982

[3] ISO Minimal BASIC. Draft DIS 6376, International Organization for Standarization Geneva, 1980

[4] Iszkowski W., Maniecki M.: Programowanie w języku BASIC PWN, Warszawa, 1982 (II wydanie)

[5] Kurtz T. E.: On the Way to Standard BASIC. BYTE, pp. 182—218, June 1982

[6] Minimal BASIC. Standard ECMA-55, European Computer Manufactures Association, Geneva, January 1978.

KONFERENCJE

Informatyka narzędziem ekonomisty

Reforma gospodarcza wymaga zmian i przewartościowań orientacji praktycznej oraz rozszerzenia warsztatu ekonomisty. W związku z tym Komisja Informatyki Zarządu Głównego PTE zorganizowała we wrześniu ub. roku ogólnopolską konferencję „Informatyka narzędziem ekonomisty". Postulaty i wnioski jej uczestników spowodowały decyzję cyklicznego organizowania konferencji poświęconych tej problematyce — tegoroczna będzie więc drugą z kolei.

Przedmiotem konferencji będą problemy wykorzystania informatyki jako narzędzia pracy ekonomisty w przedsiębiorstwach, problemy informatyzacji planowania, analiz ekonomicznych i rachunkowości, a także aspekty prawne i ekonomiczne zastosowania informatyki. Zaprezentowane zostaną gotowe systemy informatyczne dla ekonomistów w przedsiębiorstwach oraz najnowsze kierunki rozwoju w tym zakresie.

Celem konferencji jest wymiana doświadczeń w zakresie zastosowania informatyki w pracy ekonomisty, prezentacja jej nowych możliwości oraz wykorzystania dotychczasowego dorobku w nowych warunkach działalności przedsiębiorstw i funkcjonowania gospodarki narodowej.

Konferencja odbędzie się w IV kwartale br. w Warszawie. Jest ona adresowana do kadry kierowniczej, służb ekonomicznych i organizatorskich przedsiębiorstw i podstawowych jednostek gospodarczych oraz naukowców pracujących w obszarze problemowym konferencji.

W trakcie konferencji organizowane będą pokazy krajowego sprzętu oraz oprogramowania użytkowego, ukierunkowanego na obsługę i wspomaganie ekonomisty.

Orientacyjna tematyka konferencji:

- wybrane systemy dla ekonomistów w przedsiębiorstwach
  - — ewidencja podstawowa
  - — wspomaganie prac biurowych
  - — wspomaganie rachunkowości
  - — wspomaganie analiz ekonomicznych
  - — wspomaganie planowania
- obsługa informacyjna przedsiębiorstw
  - — informacja statystyczna
  - — informacja naukowo-techniczna
  - — informacja rynkowa
- aspekty prawne i ekonomiczne wdrażania systemów informatycznych
- nowe dziedziny i narzędzie w zastosowaniach informatyki
  - — doświadczenia zagraniczne
  - — wyniki badań
  - — nowe rozwiązania.

Koszt uczestnictwa w konferencji łącznie z materiałami (zbiór referatów) — 1250 zł od osoby (1000 zł — członkowie wspierający PTE).

Zgłoszenia należy kierować pod adresem: Biuro Zarządu Głównego PTE, ul. Nowy Świat 49, 00-042 Warszawa.

HANNA OKTABA
Instytut Informatyki
Uniwersytet Warszawski

Klasy w LOGLANIE

Programowanie strukturalne jest jednym z haseł popularyzowanym od przeszło dziesięciu lat [1], nakłaniającym do pisania czytelnych programów, skomponowanych z fragmentów układających się w przejrzystą całość. Uzyskanie odpowiedniej strukturalizacji programu w językach typu asembler czy FORTRAN jest praktycznie nieosiągalne, szczególnie jeśli program wykorzystuje złożone struktury danych. Jednym z kierunków rozwoju języków programowania było poszukiwanie narzędzi do modularnego wyrażania zarówno zestawów danych, jak i zestawów instrukcji operujących na tych danych. Najbardziej udaną — jak można oceniać — konstrukcją rozwiązującą ten problem jest konstrukcja klasy wprowadzana po raz pierwszy w języku SIMULA 67 [2, 3].

Klasa jest rodzajem modułu syntaktycznego, który — najogólniej mówiąc — łączy cechy rekordu i procedury. Z jednej strony — klasa definiuje pewien obszar danych (tak jak rekord), z drugiej natomiast — ma strukturę deklaracji, taką jak procedura, tzn. może posiadać parametry, zawierać lokalne deklaracje oraz listę instrukcji.

Deklaracja modułu syntaktycznego klasy jest wzorcem, zwane o b i e k t a m i. Tworzenie obiektów następuje w sposób dynamiczny, w trakcie realizacji instrukcji programu. Z utworzeniem obiektu związane jest przekazanie parametrów aktualnych oraz wykonanie instrukcji zapisanych w deklaracji klasy. Różnica między obiektem klasy a dynamiczną jednostką wołanej procedury polega na tym, że obiekt, po wykonaniu swoich instrukcji, pozostaje nadal dostępny w programie jako struktura danych (nie ginie, tak jak ma to miejsce w przypadku procedur).

Przedstawione powyżej cechy klas powinny w wystarczający sposób uzasadnić wybór, jakiego dokonano opracowując koncepcję języka LOGLAN [4]. Uznano mianowicie klasy za jeden z najbardziej celnych sposobów strukturalizacji programów i postanowiono uznać klasę, obok bloku i procedury, za podstawowy moduł języka LOGLAN.

Aby przekonać Czytelników o przydatności i „urodzie" klas, przedstawiony zostanie sposób deklarowania klas w LOGLANIE, tworzenia ich obiektów oraz dokonywania operacji na obiektach jako strukturach danych. Posłuży do tego prosty z konieczności przykład, ale — miejmy nadzieję — pozwoli on Czytelnikowi zrozumieć podstawowe idee i wyobrazić sobie bardziej zaawansowane zastosowania.

DEKLARACJA KLAS

Deklaracja klasy w LOGLANIE przyjmuje następującą postać:

```
unit A: class (List-par-form);
        Dekl — lok;
        begin
          Instr;
        end A;
```

gdzie: **A** — jest nazwą klasy; **List-par-form** — listą parametrów formalnych; **Dekl-lok** — deklaracje lokalnych wielkości; **Instr** — lista instrukcji klasy.

Listy parametrów formalnych, deklaracji lokalnych czy instrukcji mogą być oczywiście puste. Wśród deklaracji lokalnych mogą pojawić się, oprócz zmiennych, również procedury, funkcje, bądź inne klasy.

Jako przykład rozważmy następującą deklarację klasy:

```
unit ułamek: class (licz, mian: integer);
  begin
    if mian=0 then ERROR (*mianownik równy zero*) else
      if licz≠0 then (*próbujemy skrócić ułamek*)
        block
          var p,q: integer;
          begin
            p:=abs(licz); q:=abs(mian);
            while p≠q do (*szukanie największego wspólnego dzielnika*)
              if p>q then p:= p—q
                     else q: = q—p
              fi
                          od;  (*p = q = uwd(licz,mian)*)
            (*skrócenie ułamka*)
            if p≠1 then licz:=licz/p;
                        mian:=mian/p fi;
          end
      fi
  fi
  end ułamek;
```

Klasa **ułamek** opisuje liczbę wymierną reprezentowaną przez parę liczb całkowitych (**licz, mian**). Lista instrukcji umieszczona w treści klasy ma rozpoznać, czy jest to dobrze określona liczba wymierna (**mian≠0**) oraz doprowadzić ułamek do postaci nieskracalnej. Jeżeli mianownik jest równy zeru, to powinien być zasygnalizowany błąd. W powyższej deklaracji sytuację tę zamarkowano napisem ERROR; w kompletnej wersji deklaracji klasy **ułamek** powinno tu nastąpić zgłoszenie sygnału o wystąpieniu sytuacji wyjątkowej w programie.

Aby jednak nie zagłębiać się w metody obsługi sytuacji wyjątkowych w LOGLANIE, sprawa ta została w omawianym przykładzie pominięta. Doprowadzenie ułamka do postaci nieskracalnej realizowane jest w lokalnym bloku klasy **ułamek** i polega na obliczeniu algorytmem Euklidesa największego wspólnego dzielnika licznika i mianownika, a następnie dokonaniu odpowiedniego dzielenia.

OBIEKTY KLAS

Deklaracja klasy jest jedynie wzorcem syntaktycznym, według którego tworzone są dynamiczne jednostki, czyli — obiekty klas. Wyrażenie umożliwiające tworzenie obiektów, zwane również g e n e r a t o r e m obiektów, ma ogólną postać:

```
new A(List-par-akt)
```

gdzie: **A** jest nazwą klasy, której obiekt jest tworzony; **List-par-akt** — jest listą parametrów aktualnych. Obliczenie wartości takiego wyrażenia składa się z następujących etapów:

- rezerwowane jest nowe pole pamięci dla parametrów i wielkości lokalnych zadeklarowanych w klasie **A** — jest to pole obiektu

● obliczana jest wartość parametrów aktualnych i przekazywana do zarezerwowanego pola, zgodnie z przyjętymi regułami przekazywania parametrów

● wykonywane są instrukcje umieszczone w deklaracji klasy **A** dla których pole obiektu jest lokalną strukturą danych.

Wartością wyrażenia:

new A (...)

jest referencja (adres) pola obiektu. Wartość ta jest uzyskiwana po zrealizowaniu wymienionych wyżej kroków i może być przechowywana przez zmienne typu **A**. Deklaracja zmiennych typu **A** ma postać:

var X : A

Mając deklarację klasy **A** i zmiennej **X** typu **A** możemy napisać kompletną instrukcję przypisania:

X := **new** A(List-par-akt);

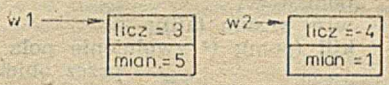
w wyniku której zostanie utworzony nowy obiekt klasy **A**; zmienna **X** będzie wskazywała adres utworzonego obiektu. Zauważmy, że po utworzeniu obiektu (czyli między innymi po wykonaniu instrukcji klasy) pole obiektu nie ginie (tak jak w przypadku procedur), stanowi natomiast pewną strukturę danych dostępną w dalszej części programu dzięki zmiennej wskazującej to pole.

Kontynuując nasz przykład, rozpatrzmy następujący fragment programu:

```
var w1, w2 : ułamek;
begin
w1 := new ułamek(9,15);
w2 := new ułamek(—8,2);
...
end
```

W wyniku wykonania obu instrukcji przypisania powstaną dwa obiekty klasy **ułamek**, na które wskazują odpowiednio zmienne **w1** i **w2** (rys. 1).

Wartości parametrów obu obiektów zostały zmienione w wyniku wykonania instrukcji w trakcie tworzenia obiektów.



Rys. 1. Obiekty klasy „ułamek"

DOSTĘP DO ATRYBUTÓW OBIEKTU

Wielkości lokalnie zadeklarowane w klasie lub umieszczone na liście parametrów nazywamy atrybutami klasy. Pojęcie to przenosi się na poszczególne obiekty klas (w naszym przykładzie powiemy, że **licz** i **mian** są atrybutami obiektów **w1** i **w2**. Ponieważ obiekty klas nie giną po ich utworzeniu, powstaje naturalne pytanie, w jaki sposób można korzystać z atrybutów istniejących obiektów. Stosuje się metodę odległego dostępu (zwanego też dostępem „kropkowanym"), wprowadzoną w SIMULI 67, a następnie wykorzystaną w PASCALU w dostępie do pól rekordów.

Metoda ta polega na dotarciu z zewnątrz do atrybutów obiektu, w celu odczytania lub zmienienia ich wartości bądź wywołania — jeżeli są to procedury lub funkcje. Dotarcie do obiektu następuje oczywiście poprzez zmienną wskazującą obiekt. Umieszczenie po tej zmiennej kropki, a następnie wypisanie nazwy atrybutu, identyfikuje jednoznacznie atrybut, na którym możemy dokonać zamierzonej operacji.

Dla przykładu — wykonanie instrukcji:

w1.licz:= w1.licz+1;

spowoduje zmianę wartości atrybutu **licz** (**licz**=4) w obiekcie **w1**.

Łatwo zauważyć, że dostęp do atrybutów będących zmiennymi, pozwala na dokładnie takie samo manipulowanie obiektami, jak ma to miejsce w przypadku dostępu do pól rekordów w PASCALU. Przypadek ten uznamy więc za mniej interesujący, postaramy się zaś wyjaśnić Czytelnikowi zalety dostępu do atrybutów, które są procedurami lub funkcjami.

W tym celu zmodyfikujemy nieco nasz przykład. Wyobraźmy sobie, że chcemy wykonywać operacje dodawania i mnożenia ułamków. Oczywiście, operacje te można zdefiniować jako funkcje „na zewnątrz" klasy **ułamek**. Dla przykładu przytoczymy deklarację funkcji sumującej ułamki.

```
unit sum : function (P,Q: ułamek) : ułamek;
           begin
             result:= new ułamek (P.licz*Q.mian
                                  + P.mian*Q.licz,
                                  P.mian*Q.mian);
           end sum;
```

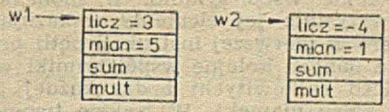
Można jednakże operacje na ułamkach zaleklarować jako atrybuty klasy definiującej ułamki. Rozważmy zmodyfikowaną wersję klasy **ułamek**:

```
unit ułamek1: class (licz,mian:integer);
        unit sum : function (P:ułamek1) : ułamek1;
                   begin
                                 result:=new ułamek1
                                       (licz*P.mian +
                                        mian*P.licz,
                                        mian*P.mian);
        end sum;
        unit mult: function (P:ułamek1): ułamek1;
                   begin
                                 result:=newułamek1
                                       (licz*P.licz,
                                        mian*P.mian);
                   end mult:
        begin
          (*lista instrukcji dokładnie taka
            sama jak w klasie ułamek * )
        end ułamek1;
```

Atrybutami każdego obiektu klasy **ułamek1** będą nie tylko parametry **licz** i **mian**, ale również funkcje **sum** i **mult**. Załóżmy, że zmienne **w1**, **w2**, **S** są typu **ułamek1**. Wykonanie instrukcji:

```
w1:= new ułamek1 (9,15);
w2:= new ułamek1 (—8,2);
```

spowoduje powstanie, tak jak uprzednio, dwóch obiektów (rys. 2).



Rys. 2. Obiekty klasy „ułamek1"

Każdy z tych obiektów dysponuje „prywatną" funkcją **sum** i **mult**. „Prywatność" tych funkcji względem obiektów polega na tym, że funkcje te w swej treści korzystają z atrybutów **licz** i **mian** danego obiektu — tak jak z wielkości lokalnych, natomiast wartość drugiego ułamka, który ma być dodany (pomnożony) pobierają z zewnątrz obiektu — poprzez parametr. Dodajmy — dla przykładu — ułamek **w1** do **w2**, korzystając z atrybutu **sum** obiektu **w1**.

S:= w1.sum(w2);

Wykonanie tej instrukcji polega na wywołaniu funkcji **sum** w otoczeniu obiektu **w1**, tzn. wartości zmiennych **licz** i **mian**, występujące w treści funkcji **sum** będą miały takie wartości jak w obiekcie **w1**, natomiast atrybuty obiektu **w2** będą dostępne „przez kropkę".

Podstawową zaletą deklarowania operacji na strukturze danych wewnątrz deklaracji klasy opisującej tę strukturę jest uzyskanie bezpośredniego dostępu do elementów stru-

ktury. Mniej zauważalną cechą, ale wartą odnotowania, jest również możliwość uzyskania infiksowego zapisu operacji dwuargumentowych. Zapis: **S:= w1.sum(w2);** jest bardziej zbliżony do naturalnego: **S:= w1 + w2,** aniżeli **S:= sum (w1, w2),** który należałoby stosować w przypadku operacji deklarowanych na zewnątrz klasy.

Jako podsumowanie niniejszych rozważań przytoczymy fragment programu, w którym zostanie wykorzystana struktura danych zadeklarowana w klasie **ułamek1.**

Załóżmy, że dany jest wielomian o współczynnikach wymiernych:

$$w(x) = a_0x^n + a_1x^{n-1} + \ldots + a_{n-1}x + a_n$$

Napiszemy fragment programu obliczający wartość tego wielomianu w punkcie wymiernym **b.** Posłużymy się schematem Hornera obliczania wartości wielomianu:

$$w(b) = (\ldots((a_0 * b + a_1) * b + a_2) \ldots + a_{n-1}) * b + a_n$$

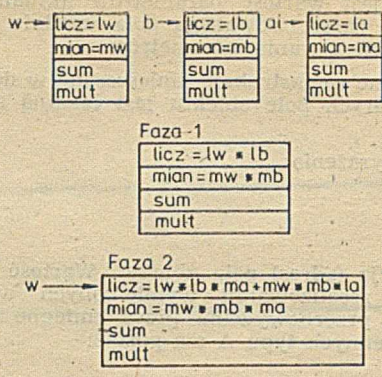
Ze względu na oszczędność miejsca nie będziemy przytaczali ponownie całej treści klasy **ułamek1,** zaznaczymy tylko miejsce jej wystąpienia.

```
program Horner;
  unit ułamek1: class ... end ułamek1;
  var a: arrayof ułamek1, w,b: ułamek1,  i,l,m,n:  integer;
  begin
    read (n);
    array a dim (0 : n); (* utworzenie tablicy współczyn-
                                                  ników *)
    for i:=0 to n do (* wczytanie współczynników wielo-
                mianu jako par liczb całkowitych i utworze-
                        nie obiektów dla współczynników *)
      read (l,m);
    a(i) : = new ułamek1 (l,m);
                od;
    (* wczytanie i utworzenie punktu wymiernego b *)
read (l,m); b: = new ułamek(l,m);
    (* w — zmienna pomocnicza do zliczania wartości
      wielomianu, początkowo równa zero *)
    w:= new ułamek1(0,1);
    (* obliczanie wartości wielomianu w punkcie b sche-
    matu Hornera *)
    for i:=0 to n do
        w:= w.mult (b).sum(a(i)); (* w:= w*b+a(i)*)
        od;
    (* w — jest liczbą wymierną będącą wartością wielo-
                                        mianu w punkcie b*)
    . . .
end
```

Pierwsze dwie instrukcje programu to wczytanie rozmiaru wielomianu, a następnie utworzenie (dynamiczne) tablicy współczynników **a.** Wartościami wpisanymi w tablicę **a** będą referencje obiektów klasy **ułamek1** reprezentujących kolejne współczynniki. W momencie utworzenia tablicy **a** wszystkie jej elementy wskazują obiekt pusty **none.** Dopiero w pierwszej instrukcji pętli programu, wczytywane są z danych kolejne współczynniki wielomianu (jako pary liczb całkowitych) i dla każdej pary tworzony jest obiekt klasy **ułamek1.** W trakcie tworzenia obiektów, współczynniki sprowadzane są do postaci ułamków nieskracalnych oraz adres utworzonego obiektu przekazywany jest jako wartość zmiennej a(i).

Kolejne instrukcje powodują wygenerowanie obiektu **b** reprezentującego punkt, w którym liczona będzie wartość wielomianu, oraz nadanie wartości początkowej zmiennej **w** (**w = (0,1)** — ułamek reprezentujący zero), na której będzie obliczana wartość wielomianu. Ostatnia instrukcja pętli w programie zlicza wartość wielomianu w punkcie **b** według schematu Hornera. Realizacja wyrażenia **w.mult(b).sum(a(i))** przebiega w następujących fazach: (1) w obiekcie klasy **ułamek1,** wskazywanym przez **w,** zostaje wywołana funkcja **mult,** która tworzy nowy obiekt klasy **ułamek1.** Obiekt ten odpowiada iloczynowi ułamków reprezentowanych przez **w** i **b** (2), w tym nowo utworzonym obiekcie zostaje wywołana funkcja **sum** z parametrem **a(i),** która powoduje utworzenie kolejnego obiektu klasy **ułamek1,** będącego sumą **w.mult(b)** oraz **a(i).**

Adres obiektu klasy **ułamek1** utworzony w fazie (2), stanowi końcową wartość zmiennej **w** po wykonaniu instrukcji **w:=w.mult(b).sum(a(i)).** Rysunek 3 przedstawia graficznie kolejne fazy realizacji tej instrukcji przypisania.



Rys. 3. Graficzny obraz realizacji instrukcji w:=w.mult(b).sum(a(i)). Faza (1) — obiekt powstały w wyniku obliczenia wyrakenia w.mult(b); faza (2) — obiekt wynikający z wywołania funkcji sum(a(i)) w obiekcie powstałym w fazie (1)

Można zauważyć, że w każdym kolejnym kroku instrukcji pętli powstają dwa nowe obiekty klasy **ułamek1.** Obiekty te pełnią rolę pomocniczą, wartości ich atrybutów są wykorzystywane jednorazowo w trakcie obliczeń, natomiast pola tych obiektów niepotrzebnie „zaśmiecają" pamięć w dalszej części obliczania programu. Ażeby radzić sobie z takimi sytuacjami, wprowadzono w LOGLANIE instrukcję **kill(...),** pozwalającą na zwalnianie pól niepotrzebnych obiektów. Jeżeli **A** jest zmienną wskazującą obiekt, z którego już nie chcemy korzystać, to wykonanie instrukcji **kill(A)** spowoduje zwolnienie pola zajmowanego przez obiekt **A,** wartością zaś zmiennej **A** będzie obiekt pusty **none.** W naszym programie, w ostatniej instrukcji pętli, niepotrzebnymi w dalszych obliczeniach obiektami są: „stara" wartość zmiennej **w** oraz obiekt powstały w fazie (1) obliczania wartości wyrażenia (rys. 3). Zmodyfikowana wersja instrukcji pętli, zwalniająca pola niepotrzebnych obiektów, może przyjmować postać:

```
for i: =0 to n do
                pom:= w; (*pom — zmienna pomocnicza ty-
                pu ułamek1*)
                w:= w.mult(b);
                kill (pom); (* zwolnienie pola zajmowanego
                przez „stary" obiekt w *)
                pom:= w;
                w:= w.sum (a(i));
                kill (pom); (* zwolnienie pola zajmowanego
                                przez obiekt w.mult(b)*)
            od
```

Na prostym przykładzie liczb wymiernych i operacji na nich pokazana została przydatność klas do modularnego definiowania struktur danych oraz wygodnego manipulowania nimi. Faktem jest jednak, że podobną strukturalizację można osiągnąć innymi środkami, np. za pomocą rekordów i procedur pascalowych. Prawdziwa siła klas ujawnia się dopiero przy omówieniu koncepcji łączenia modułów klas w większe całości, zwana p r e f i k s o w a n i e m k l a s.

Temat ten zostanie omówiony w jednym z najbliższych numerów INFORMATYKI.

Bardzo istotnym zastosowaniem klas jest również możliwość definiowania za ich pomocą współprogramów, czyli takich składowych programu, które mogą współpracować ze sobą w sposób równoprawny. To także znajdzie niebawem wyraz na łamach INFORMATYKI.

LITERATURA

[1] Dahl O.J., Dijkstra E.W., Hoare C.A.R.: Structured programming. London, Academic Press, 1972

[2] Dahl O.J., Myhrhaug B., Nygaard K.: Simula-67 Commone Base Language. Norwegian Computing Center, 1970

[3] Oktaba H., Ratajczak W.: Simula 67, WNT, 1980

[4] Report on the programming language LOGLAN. Praca zbiorowa, maszynopis, Instytut Informatyki UW, 1982.

JACEK OWCZARCZYK
Warszawa

Parametry użytkowe wybranych rastrowych systemów graficznych

Systemy graficzne o organizacji rastrowej, opisane w poprzednim numerze INFORMATYKI, dominują na rynku komputerowego sprzętu graficznego. Coraz mniej firm produkuje dziś grafoskopy. Również firma TEKTRONIX, producent znanych terminali graficznych z lampą pamięciową, wprowadza w nowych modelach organizację rastrową.

Mnogość firm uniemożliwia praktycznie zbiorcze porównanie parametrów wszystkich systemów graficznych o organizacji rastrowej. Niniejszy artykuł jest próbą przedstawienia systemów produkowanych przez czołowe firmy amerykańskie. Wybór ten nie jest przypadkowy. Szacuje się, że łączne wpływy firm amerykańskich ze sprzedaży komputerowego sprzętu graficznego wyniosą w bieżącym roku 4,5 mld dolarów [1]. W trakcie opracowywania materiału uwzględniono systemy graficzne produkowane przez firmy: COMTAL [2], DE ANZA SYSTEMS [4, 7, 16], GENISCO [3, 10, 14], GRINNELL SYSTEMS [5], INTERNATIONAL IMAGING SYSTEMS (I²S) [11, 12], MEGATEK [18, 19], RAMTEK [13], oraz MICRO CONSULTANTS [8, 9] czołowego producenta systemów graficznych na rynku brytyjskim.

PODSTAWOWE PARAMETRY

W tabeli przedstawiono podstawowe parametry analizowanych systemów graficznych. Najbardziej powszechne wymiary r a s t r a to 512 elementów obrazu (ang. pixels) na jedną linię i 512 linii. Należy podkreślić, że dotyczy to głównie systemów produkowanych na rynek europejski, gdzie obowiązują standardy telewizyjne 625 linii (PAL, SECAM). W USA, z uwagi na odmienny standard telewizyjny (NTSC) 525 linii, wyświetlany raster ma format 512 × × 480. Są to wymiary rastrów umożliwiające stosowanie typowych monitorów telewizyjnych.

Sposób organizacji pamięci obrazowej wiąże się ściśle z przeznaczeniem systemu graficznego. W systemach wykorzystywanych do cyfrowego przetwarzania obrazów każdemu elementowi obrazu przyporządkowana jest grupa bitów w pamięci obrazowej. W systemach monochromatycznych najczęściej przeznacza się osiem bitów, istnieje więc możliwość kodowania 256 różnych poziomów szarości. W systemach generujących obrazy barwne dla każdego z podstawowych kolorów przeznacza się osiem bitów (rzadziej cztery).

W systemach przeznaczonych głównie do wyświetlania rysunków złożonych z linii, a także do tworzenia synte-

Zestawienie podstawowych parametrów systemów graficznych o organizacji rastrowej

Producent (kraj)	Model	Wymiary rastra (liczba elementów/linię × liczba linii)	Organizacja pamięci obrazowej (H × V × B × liczba kanałów)	Procesor	Zastosowanie
COMTAL (USA)	30 SER	512×512	512×512×8×3	—	przetwarzanie obrazów
	65 SER	1024×1024	1024×1024×8×3	—	
	Vision One/10	512×512	512×512×8×4	LSI-11/02	
DE ANZA SYSTEMS (USA)	VISACOM/23	512×480	512×480×12×1	LSI-11/23	przetwarzanie obrazów
	IP 8500	512×512	512×512×8×20	LSI-11	
	IP 5532	512×512	512×512×8×4	LSI-11	
GENISCO (USA)	G 1000	1024×792	1024×1024×1×1	Z8001	grafika komputerowa
	GCT 3000	1280×1024	1280×1024×24×1	specjalizowany — złożony z mikroprocesorów modułowych	grafika komputerowa (przetwarzanie obrazów)
GRINNELL SYSTEMS (USA)	GMR 274	512×512	512×512×8×3	—	przetwarzanie obrazów
	GMR 275	512×512	512×512×8×5	—	
INTERNATIONAL IMAGING SYSTEMS (I²S) (USA)	75	512×512	512×512×8×16	LSI-11/23	przetwarzanie obrazów
	70	512×512	512×512×8×12	LSI-11	
MEGATEK (USA)	6240	512×512	512×512×1×1	specjalizowany — złożony z mikroprocesorów modułowych	grafika komputerowa
	7255	1024×1024	4096×4096×3×1		
MICRO CONSULTANTS (Wielka Brytania)	INTELLECT 100	512×512	512×512×8×1	LSI-11	przetwarzanie obrazów
	INTELLECT 200	512×512	512×512×8×12	LSI-11/23	
RAMTEK (USA)	9400	1280×1024	1280×1024×16×1	specjalizowany — złożony z mikroprocesorów modułowych	grafika komputerowa (przetwarzanie obrazów)

tycznych obrazów (w ostatniej rubryce tabeli: grafika komputerowa) pamięć obrazowa składa się z jednej (systemy monochromatyczne) lub kilku (systemy kolorowe) jednobitowych płaszczyzn. Dodatkowa pamięć obrazowa o podobnej funkcji i organizacji występuje również w systemach przetwarzania obrazów; służy ona do rejestracji opisów alfanumerycznych i wykresów graficznych, które mogą być nakładane (ang. overlay) na wyświetlany obraz.

W rozbudowanych systemach graficznych istnieje możliwość przetwarzania kilku obrazów równolegle. Każdy kanał ma wówczas oddzielną pamięć obrazową. Podane w tabeli wielkości pamięci obrazowej dotyczą maksymalnych konfiguracji systemów. W niektórych systemach organizacja pamięci obrazowej może być zmieniana programowo (np. z $512 \times 512 \times 8 \times 4$ na $1024 \times 1024 \times 8 \times 1$). Czas dostępu do pamięci obrazowej wynosi od 0,8 μs/element obrazu (system 75) do 1,5 μs (systemy GMR 274 i GMR 275).

W konfiguracji aktualnie produkowanych systemów graficznych bardzo często znajduje się procesor, którego głównymi zadaniami są: nadzorowanie komunikacji z zewnętrznym systemem komputerowym, realizacja niektórych programów użytkowych, obsługa graficznych urządzeń wejściowych. W systemach grafiki komputerowej pełni on funkcję procesora graficznego, wykonującego rozkazy programów graficznych.

W systemach przetwarzania obrazów najczęściej używany jest jeden z mikrokomputerów serii LSI-11, produkowanych przez DIGITAL EQUIPMENT CORP. Atrakcyjność tego procesora polega na tym, że zachowuje on wiele cech funkcjonalnych (lista rozkazów, sposoby adresowania) popularnych komputerów serii PDP-11 i umożliwia stosowanie typowych dla tych komputerów systemów operacyjnych: RT11 (LSI-11, LSI-11/02, LSI-11/23) lub RSX11M (LSI-11/23). W szczególnych przypadkach procesory te umożliwiają konstrukcję całkowicie niezależnych systemów komputerowych.

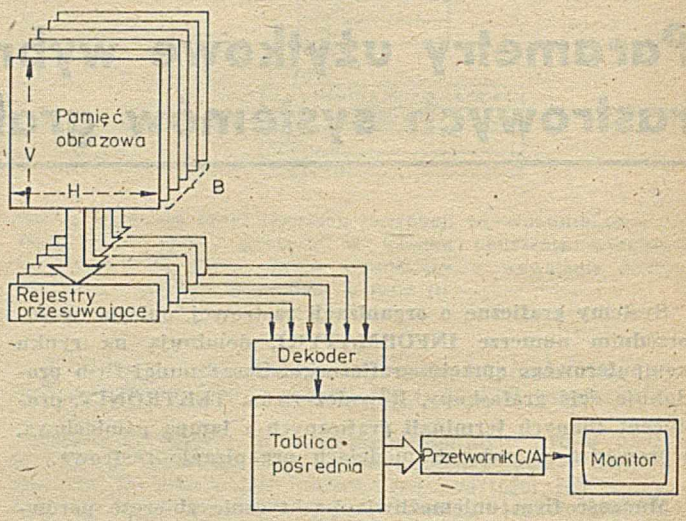
W systemach grafiki komputerowej typowe mikroprocesory są zbyt wolne (ze względu na rozbudowane schematy adresowania, nie wymagane w zastosowaniach graficznych), stosuje się więc specjalizowane procesory oparte na mikroprocesorach modułowych (ang. bit slice).

TYPOWE UKŁADY

W celu zwiększenia szybkości obliczeń, niektóre funkcje w systemach graficznych są realizowane sprzętowo. W systemach przetwarzania obrazów są to zawsze operacje manipulowania wyświetlanym obrazem, takie jak: powiększanie wybranej części obrazu (ang. zoom), przemieszczenie obrazu na ekranie monitora (ang. scroll) lub wyświetlanie kilku fragmentów różnych obrazów jednocześnie na ekranie (ang. split screen). Inne funkcje — jak generowanie wektorów czy znaków alfanumerycznych — mogą być realizowane w tych systemach sprzętowo lub programowo.

W systemach grafiki komputerowej generowanie wektorów odbywa się zawsze sprzętowo. W systemie 7255 (MEGATEK) do generowania wektorów, a następnie do zmiany struktury danych na rastrową, używany jest specjalny układ Digital Vector Generator. Konwersja danych odbywa się z szybkością 100 ns/element obrazu. W tym samym systemie możliwa jest sprzętowa realizacja takich operacji, jak: kadrowanie (ang. clipping) tworzonego rysunku do zadanej wielkości rastra oraz podstawowe transformacje geometryczne wyświetlanego rysunku (skalowanie, obrót, przesunięcie). System 9400 (RAMTEK) umożliwia sprzętowe generowanie wektorów z szybkością 16 tys. wektorów na sekundę.

Tablica pośrednia (ang. Look-Up-Table) jest układem, w który wyposażone są wszystkie systemy graficzne o organizacji rastrowej (rys.). Zawartość kolejnych bajtów pobieranych z pamięci obrazowej przez układ formowania sygnału wizyjnego jest traktowana jako adres elementu tablicy pośredniej, a informacja zapisana pod tym adresem służy do sterowania katodami monitorów. Tablica pośrednia ma tyle elementów, ile różnych wartości można przyporządkować pojedynczemu elementowi obrazu. Jej szerokość (w bitach) określa, ile różnych poziomów szarości lub barw może być jednocześnie wyświetlanych na ekranie monitora. Zawartość tablicy pośredniej może być wpisywana programowo. Jest to niezbędny układ w systemach przetwarzania obrazów. Większość wykonywanych tam operacji może być realizowana przez odpowiednie wypełnianie tablicy pośredniej. W systemach wielokanałowych każdy kanał ma oddzielną tablicę pośrednią.



Zasada działania tablicy pośredniej

Systemy przetwarzania obrazów wyposażone są w moduł umożliwiający cyfrowanie i umieszczanie w pamięci obrazowej zewnętrznego sygnału wizyjnego (z kamery telewizyjnej, odbiornika telewizyjnego lub magnetowidu). Sygnał cyfrowany jest zwykle z częstotliwością 10 MHz i dokładnością ośmiu bitów, co umożliwia cyfrowanie poszczególnych kadrów w czasie rzeczywistym. Operacje w systemie graficznym, które realizowane są w trybie nadążnym, mogą być stosowane do otrzymanych w ten sposób danych, co umożliwia przetwarzanie obrazów na bieżąco. Wówczas moduł ten traktowany jest umownie jako jeden z kanałów. W USA stosowany jest standard sygnału wizyjnego RS170, w Europie CCIR (1 Vpp).

W niektórych systemach (GMR 274, GMR 275, INTELLECT 200), po zamianie sygnału wizyjnego na sygnały cyfrowe, mogą być one wstępnie przetworzone przed przesłaniem do pamięci obrazowej. W systemach tych istnieje również możliwość odejmowania cyfrowego odpowiednika sygnału wizyjnego od zawartości pamięci obrazowej.

SPRZĘGI

Komunikacja systemów graficznych z innymi systemami komputerowymi dokonuje się przez sprzęgi szeregowe lub równoległe. W systemach przetwarzania obrazów, z uwagi na znaczne ilości transmitowanych danych, stosuje się sprzęgi równoległe. W analizowanych systemach graficznych oferowane są sprzęgi równoległe dla wszystkich najbardziej popularnych systemów komputerowych, pośród których są zawsze komputery serii PDP-11 (DEC), NOVA (DG), HP3000 (HP). Systemy grafiki komputerowej wyposażone są zawsze w sprzęg szeregowy typu RS 232C (który jest w zasadzie zgodny z europejskim standardem V24). Typową szybkością transmisji jest tu 9600 bodów.

GRAFICZNE URZĄDZENIA WEJŚCIOWE

Wszystkie aktualnie produkowane systemy graficzne umożliwiają współdziałanie użytkownika z systemem w trakcie przetwarzania informacji. Lokalizacja żądanych elementów obrazu na ekranie monitora odbywa się za pomocą różnych urządzeń wejściowych: manipulatorów kulowych (ang. trackball), manipulatorów drążkowych (ang. joystick), piór świetlnych lub klawiatur funkcyjnych. Sterują one znacznikiem (ang. cursor) wyświetlanym na ekranie monitora. W wielu systemach kształt, wielkość i kolor znacznika mogą być programowane. Położenie znacznika na ekranie wyznaczają rejestry X, Y, których zawartość może być ustawiana programowo lub za pomocą urządzeń wej-

ściowych. W niektórych systemach generowane są niezależnie dwa (GMR 274, VISACOM 23) lub cztery (GMR 275) znaczniki.

W systemach grafiki komputerowej najczęściej, prawdopodobnie ze względu na tradycję grafoskopów, używane są pióra świetlne. W systemach przetwarzania obrazów przeważnie stosuje się manipulatory kulowe.

UKŁADY SPECJALNE

W celu dalszego zmniejszenia czasu przetwarzania danych w wielu systemach graficznych pewne typowe operacje realizowane są sprzętowo. Na przykład — w systemach GMR 274 i GMR 275 specjalny układ (Image Analyzer) pozwala generować histogramy częstości występowania poszczególnych poziomów szarości dla części lub całego obrazu, a także znajduje minimalną i maksymalną wartość w pamięci obrazowej. Układ o podobnych funkcjach istnieje również w systemach 75 i 70 (I^2S).

W systemie INTELLECT 200 moduł realizujący sprzętowo szybką transformatę Fouriera (FFT) pozwala zmniejszyć szybkość obliczeń dla obrazu 256×256 z 30 minut do kilku sekund [15]. Inne urządzenie tego samego systemu — tzw. Two Dimensional Correlator, złożone z czterech niezależnych, pracujących równolegle procesorów — umożliwia bardzo szybką (40 MHz) realizację rachunku macierzowego przy takich operacjach, jak: filtrowanie, detekcja brzegów, redukcja zakłóceń (szumów).

W systemach przetwarzania obrazów większość typowych operacji dokonuje się przez odpowiednie wypełnianie tablicy pośredniej przy niezmienionych wartościach w pamięci obrazowej. W wielu systemach istnieje możliwość zwrotnego przesłania przetworzonych danych (z wyjść tablicy pośredniej) do pamięci obrazowej. Umożliwia to iteracyjne przetwarzanie obrazów. W systemie 70 (I^2S) specjalne rejestry (akumulator) służą do przechowywania danych przed przesłaniem do pamięci obrazowej, jednostka arytmetyczno-logiczna umożliwia wykonanie jednej z 48 operacji pomiędzy danymi z akumulatora i danymi z tablicy pośredniej.

OPROGRAMOWANIE

W oferowanym przez producentów systemów graficznych oprogramowaniu można wyodrębnić następujące elementy:

— program obsługi, umożliwiający komunikację z systemem graficznym; jest on częścią systemu operacyjnego systemu komputerowego, w skład którego wchodzi system graficzny

— oprogramowanie użytkowe

— programy diagnostyczne

— programy specjalistyczne.

W systemach przetwarzania obrazów oprogramowanie użytkowe obejmuje bibliotekę programów, którą można podzielić na następujące grupy tematyczne: zarządzanie zbiorem obrazów, przesyłanie obrazów między poszczególnymi układami systemu, operacje arytmetyczne na obrazach, opisy statystyczne obrazów (histogramy, przekroje), transformacje obrazów, manipulowanie obrazami, klasyfikacja obrazów. Oprogramowanie użytkowe SYSTEM 500 dla systemu graficznego 70 (I^2S) składa się z 100 tys. linii programu napisanego w języku FORTRAN.

W skład oprogramowania użytkowego sytemów grafiki komputerowej wchodzą zazwyczaj następujące programy: transformacje geometryczne (przesunięcia, skalowanie, obroty) wyświetlanego rysunku, kadrowanie, transformacje obiektów w przestrzeni trójwymiarowej, wypełnianie zadanych figur geometrycznych kolorem, rzuty perspektywiczne. Przykładem oprogramowania użytkowego obejmującego wszystkie wymienione wyżej programy są programy TEMPLATE i WAND dla systemu graficznego 7255 (MEGATEK). Oba programy napisane są w języku FORTRAN (to zresztą ciekawe, że we wszystkich analizowanych systemach graficznych programy użytkowe zostały napisane w tym języku). W większości aktualnie produkowanych systemów grafiki komputerowej oprogramowanie jest zgodne ze standardem CORE SIGGRAPH [6].

Programy specjalistyczne obejmują procedury przetwarzania obrazów typowe w konkretnych zastosowaniach. Przykładem może być program CELLS dla krystalografii (MICRO CONSULTANTS) czy programy przetwarzania obrazów satelitarnych LANDSAT.

KOSZT SYSTEMU

Mimo że koszt wytwarzania pamięci półprzewodnikowej w ostatnich latach znacznie się obniżył, to w dalszym ciągu wielkość obrazowej pamięci w decydującym stopniu wpływa na cenę systemu. Dla przykładu system 30SER (COMTAL) z pamięcią obrazową $512 \times 512 \times 8 \times 3$ i dodatkowo $512 \times 512 \times 4$ kosztował w ubiegłym roku 33 tys. $, a system 65 SER z pamięcią obrazową $1024 \times 1024 \times 8 \times 3$ i dodatkowo $1024 \times 1024 \times 1$ kosztował 106 tys. $. W tym samym czasie system GCT 3000 (GENISCO) z pamięcią obrazową $1024 \times 1024 \times 3$ kosztował 24 tys. $. Ceny te obejmują tylko podstawową konfigurację systemu i nie zawierają kosztów monitora, graficznych urządzeń wejściowych, układów specjalnych i oprogramowania.

* * *

W hierarchii komputerowego sprzętu graficznego analizowane systemy można zaliczyć do systemów średnich i dużych. Zajmują one środkowe pozycje — pomiędzy terminalami graficznymi (o mniejszych możliwościach, ale znacznie mniej kosztownymi) a rozbudowanymi systemami graficznymi symulatorów lotów, których koszt sięga kilku mln $.

Wybór konkretnego modelu systemu graficznego powinien być przy zakupie poprzedzony gruntowną analizą jego parametrów, określających jego przydatność do realizacji zamierzonych celów. Zwłaszcza analiza kosztów często uzasadnia przyjęcie alternatywnego rozwiązania — opracowanie systemu graficznego we własnym zakresie.

LITERATURA

[1] Catier E.: Terminaux á écran: vers la couleur. Electronique Industrielle, nr 13, 1981

[2] COMTAL image processing: 8000-R series. COMTAL, Altadena, 1981

[3] GENISCO GCT-3000 series computer graphic display products. GENICSO Computers Corp., Costa Mesa, 1981

[4] Gnauck G.E., Hu P.C.: Image processing and geographic information systems and their application to resource management and exploration. ESL Inc., 1981.

[5] GMR 274, 275 image processing systems general description. GRINNELL Systems, San Jose, 1981

[6] Graphic Standards Planning. Committee status report ACM/SIGGRAPH. Computer Graphics, vol. 11, nr 3, 1977

[7] IDIMS interactive digital image manipulation system. ESL Inc., ESL-SR337, 1981

[8] INTELLECT 100 the desk top image processing system, MICRO CONSULTANTS Ltd, Kenley, 1982

[9] INTELLECT 200 a modular image processing system. MICRO CONSULTANTS Ltd, Kenley, 1981

[10] Jones D., Fletcher J.: Graphics and imaging system offloads host computer. Computer Design, nr 7, 1981

[11] Model 75 image processor. Product bulletin, INTERNATIONAL IMAGING SYSTEMS, Milpitas, 1982

[12] New in image processing model 70 image computer. INTERNATIONAL IMAGING SYSTEMS, Milpitas, 1981

[13] RAMTEK graphics systems: RM 9400 series. RAMTEK, Santa Clara, 1981

[14] The brilliant graphics terminal G-1000 from GENISCO. GENISCO Computers Corp., Costa Mesa, 1981

[15] Two significant advances in image processing. Systems International, nr 2, 1982

[16] VISACOM/23, visual computer system. Product description manual, DE ANZA Systems, doc. nr 800227, 1981

[17] WHIZZARD 6200 family of raster graphics systems. MEGATEK Corp., San Diego, 1981

[18] WHIZZARD 7200 family graphics engine. MEGATEK Corp. San Diego, 1981.

JERZY GRABOWSKI
Warszawa

Przegląd mikroprocesorów 16-bitowych (3)
MOTOROLA 68000

Mikroprocesor MOTOROLA 68000 jest pierwszym mikroprocesorem 16-bitowym o 32-bitowej architekturze wewnętrznej, ma 17 rejestrów 32-bitowych oraz 16-bitowy rejestr stanu. W grupie 17 rejestrów jest osiem danych, siedem rejestrów adresowych i dwa wskaźniki stosu przeznaczone dla dwóch trybów pracy procesora — normalnej i systemowej. Rejestry danych mogą być używane do operacji na liczbach 8-, 16- i 32-bitowych. Każdy z rejestrów adresowych może być stosowany jako programowy wskaźnik stosu, może przechowywać adres bazy lub być rejestrem indeksowym w operacjach na danych o różnej długości. Rejestry danych, oprócz zasadniczej funkcji przechowywania argumentów i wyników operacji arytmetyczno-logicznych, mogą także pełnić funkcję rejestrów indeksowych. Ponieważ rejestry adresowe mogą służyć także do wykonywania operacji arytmetycznych, można uważać, że omówione rejestry są uniwersalne.

Licznik rozkazów (32-bitowy) współpracuje z 24-bitową szyną adresową. Pamięć mikroprocesora nie jest segmentowana, a więc programista ma bezpośredni dostęp do 16 M bajtów pamięci, przeznaczonej na program i dane. Jedynie osiem bajtów musi być umieszczone w pamięci stałej (ROM), gdyż zawiera wektor wznowienia pracy systemu. Początkowe 1 K słów pamięci jest przeznaczone na inne wektory przerwań wewnętrznych i zewnętrznych, które mogą być umieszczane zarówno w pamięci stałej, jak i zapisywalnej (RAM).

Mikroprocesor MOTOROLA 68000 ma 16-bitową szynę danych i linie asynchronicznej kontroli wolnych urządzeń zewnętrznych sprzęganych przez układy z serii MC 6800. Urządzenia zewnętrzne mogą być obsługiwane metodą programowych zapytań i przez przerwania.

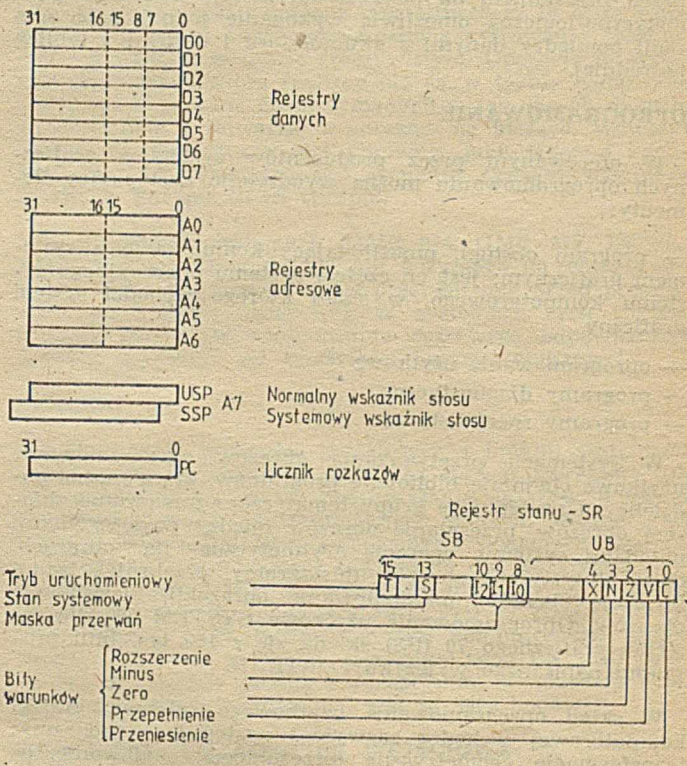
Mikroprocesor działa w dwóch trybach — normalnym i systemowym — i ma grupę rozkazów do wykorzystania tylko w trybie uprzywilejowanym. Jest więc przystosowany do pracy pod nadzorem systemu operacyjnego.

Z punktu widzenia programisty rozkazy tego mikroprocesora zostały zaprojektowane bardzo starannie. Charakteryzują się większą symetrią względem trybów adresowania, niż rozkazy mikroprocesorów omówionych uprzednio (Z8000, INTEL 8086). Przy projektowaniu układu zadbano o przystosowanie listy rozkazów do tworzenia efektywnego kodu generowanego przez translatory języków wysokiego poziomu. Mikroprocesor ma 56 grup rozkazów, które mogą działać na bitach, bajtach, 16- i 32-bitowych słowach, z wieloma trybami adresowania i w różnych wariantach, tworząc ponad 1000 elementowy zbiór wykonywalnych operacji. Powtarzając sformułowanie z prospektów reklamowych można powiedzieć, że mikroprocesor MOTOROLA 68000 został zaprojektowany przez programistów i dla programistów.

W mikroprocesorze zastosowano odwzorowanie rejestrów urządzeń zewnętrznych w obszar adresowy pamięci (ang. memory mapped i/o). Operacje zewnętrzne są inicjowane przez przesłania z (lub do) obszaru adresowego zarezerwowanego na rejestry urządzeń. W konsekwencji — zbędne jest wyodrębnianie oddzielnej grupy rozkazów wejścia-wyjścia.

REJESTRY

Rejestry mikroprocesora MOTOROLA 68000 przedstawiono na rysunku. W pierwszej grupie znajdują się rejestry danych D0, D1,...,D7. Linie przerywane dzielące rejestry na tym rysunku oznaczają możliwość wykonywania operacji na słowach 16-bitowych i bajtach. Drugę grupę stanowią rejestry adresowe A0,...,A6, które służą do wykonywania operacji 16 lub 32-bitowych. Rejestr A7 (podwójny) jest wskaźnikiem stosu dla dwóch trybów pracy procesora. Licznik rozkazów PC ma pojemność 32 bity; operacje realizowane wewnętrznie do obliczenia adresu efektywnego są wykonywane na 32 bitach, ale wyprowadzona na zewnątrz część szyny adresowej ma 24 bity.



Rejestry mikroprocesora MOTOROLA 68000

Rejestr stanu SR składa się z bajtu sytemowego (ang. user byte — UB, system byte — SB). Pierwszy bajt zawiera bity przeniesienia (C), przepełnienia (V), zera i znaku operacji (Z i N) oraz bit pełniący funkcję bitu przeniesienia przy operacjach rozszerzania znaku (X). W bajcie systemowym jest przechowywana trzybitowa maska przerwań (I0, I1, I2), która określa ograniczenie na priorytet przerwania oraz bit S określający tryb pracy procesora (normalny lub systemowy). Ustawienie bitu T powodu-

je, że po wykonaniu każdego rozkazu zgłaszane jest przerwanie wewnętrzne (ang. trace mode), które może służyć do nadzorowania działania uruchamianego programu.

TRYBY ADRESOWANIA

Mikroprocesor ma osiem podstawowych trybów adresowania, z których można wyodrębnić jeszcze dodatkowe podtryby. Łącznie istnieje 14 różnych trybów adresowania, które przedstawiono w tabeli — według podziału stosowanego przez producenta. W porównaniu z mikroprocesorami omówionymi uprzednio, MOTOROLA 68000 ma kilka nowych trybów adresowania.

Tryby adresowania mikroprocesora MOTOROLA 68000

Nazwa	Wyznaczanie adresu efektywnego	Zapis
Bezpośredni		
1. Przez rejestr danych	AE = Dn	Dn
2. Przez rejestr adresowy	AE = An	An
Pośredni		
3. Przez rejestr adresowy	AE = (An)	(An)
4. Przez rejestr adresowy z inkrementacją	AE = (An); An ← An + N	(An) +
5. Przez rejestr adresowy z dekrementacją	An ← An − N; AE = (An);	− (An)
6. Przez rejestr z wyrównaniem	AE = (An) + d16	d (An)
7. Indeksowy, przez rejestr z wyrównaniem	AE = (An) + (Ri) + + d8	d (An, Ri)
Bezwzględny (adres bezpośredni)		
8. Krótki adres	AE = (nast. słowo)	****
9. Długi adres	AE = (2 następne słowa)	********
Względny		
10. Z wyrównaniem	AE = (PC) + d16	d
11. Indeksowy z wyrównaniem	AE = (PC) + (Ri) + + d8	d (Ri)
Prosty (dane bezpośrednie)		
12. Długie dane	DANE = N następnych słów	*M* ****
13. Krótkie dane		*M* **
Niejawny		
14. Zależny od rozkazu	AE = SR, USP, SP, PC	

Oznaczenia;

AE — adres efektywny
An — n-ty rejestr adresowy
Dn — n-ty rejestr danych
SR — rejestr stanu
PC — licznik rozkazów
SP — bieżący wskaźnik stosu
Ri — rejestr danych lub adresowy jako rejestr indeksowy
USP — wskaźnik stosu normalnego
d8 — 8-bitowe wyrównanie
d16 — 16-bitowe wyrównanie
N — długość argumentu w bajtach
( ) — zawartość
← — przypisanie

Tryb pośredni przez rejestr, z inkrementacją i dekrementacją, umożliwia organizowanie danych w struktury stosowe w sposób określony przez programistę. Wskaźnikami stosów lokalnych mogą być rejestry A0, A1,...,A6. Zmiana zawartości rejestru adresowego w tych trybach adresowania jest dokonywana o 1, 2 lub 4 jednostki w zależności od kodu rozkazu, który może wytępować w różnych wariantach określających działanie na bajtach i słowach 16- lub 32-bitowych. Omawiany tryb adresowania nie jest zastrzeżony wyłącznie do przesłań, jak w przypadku opisywanych uprzednio mikroprocesorów 16-bitowych, które używały go niejawnie w rozkazach przesyłania między rejestrami a stosem (rozkazy PUSH i POP). Rozkazy mikroprocesora MOTOROLA 68000 umożliwiają więc łatwe organizowanie pętli programowych, wykonujących działania na elementach tablicy w kolejnych krokach iteracji. Obecność trybów adresowania z inkrementacją powoduje, że brak rozkazów przesyłania blokowego i innych operacji na ciągach danych nie zmniejsza istotnie operacyjnych możliwości tego mikroprocesora w stosunku do uprzednio omówionych.

Adresowanie pośrednie z wyrównaniem może dotyczyć jednego lub dwóch rejestrów (tryby 6 i 7 z tabeli), których zawartości są dodawane do wyrównania zawartego w kodzie rozkazu. W pierwszym przypadku wyrównanie może być 16-bitowe, w drugim — tylko 8-bitowe. Drugi z tych trybów adresowanja jest przydatny przy organizowaniu dostepu do tablic dwuwymiarowych. Warto zaznaczyć, że kod rozkazu zawiera informację o interpretacji rejestru adresowego, który może zawierać 16- lub 32-bitową liczbę ze znakiem. Tryby tego rodzaju tylko formalnie przypominają klasyczne adresowanie indeksowe (znane np. z opisu mikroprocesora Z8000), co wynika stąd, że pole rozkazu przeznaczone na przemieszczenie nie może pomieścić pełnego zakresu adresowego. W konsekwencji, chcąc utrzymać interpretację adresowania indeksowego, należałoby zakładać, że (dla długiego wyrównania) tablice mają początek w pierwszych 64 K słowach lub 256 słowach pamięci (dla krótkiego wyrównania). Przyjmuje się więc, że w rejestrze jest zawarty adres bazowy tablicy, a wyrównanie uważa się za indeks ustalony na czas wykonywania programu.

Adresowanie względne wystepuje w wariancie z określeniem rejestru, który pełni funkcje rejestru indeksowego (tryb 11 z tabeli). Tablice używane w tym trybie adresowania muszą być w bliskiej odległości od rozkazu (w zakresie od —128 do 127 słów względem bieżącej zawartości licznika rozkazów).

Tryby z argumentem bezpośrednim występują w kilku wariantach. Projektanci mikroprocesora pomyśleli o takim zaprojektowaniu listy rozkazów, aby dla operacji z 8-bitowym argumentem bezpośrednim oszczędzić pamięć zajmowaną przez program. Dozwolone są rozkazy z 2- lub 4-bajtowyym argumentem bezpośrednim.

Podsumowując opis trybów adresowania rozważymy wady i zalety zrealizowanego rozwiązania. Tryb indeksowy, rozumiany klasycznie, jest wyraźnie ograniczony. Można go jednak zastąpić adresowaniem pośrednim — przez rejestr. Jest to o tyle łatwe, że zbiór dostepnych rejestrów adresowych jest stosunkowo duży, a ich rezerwowanie do adresowania danych nie ogranicza możliwości wykonywania operacji arytmetyczno-logicznych z użyciem długich rejestrów arytmetycznych.

Efektywne programowanie mikroprocesora w języku asemblerowym nie jest zadaniem łatwym. Jest on predysponowany do programowania w jezykach wysokiego poziomu, gdyż standardowe sekwencje kodu, generowane przez translator, mogą być łatwo projektowane i optymalizowane ze względu na dostep do różnych złożonych struktur danych, np. języka PASCAL. Wiadomo, że rozpoczęto już projekty mające na celu realizacje języka ADA dla tego mikroprocesora, a w literaturze podkreśla się małą skuteczność programowania w języku asemblera.

LISTA ROZKAZÓW

Rozkazy mikroprocesora mogą zajmować od jednego do pięciu 16-bitowych słów pamięci. Każdy rozkaz zawiera w pierwszymy słowie kod określający rodzaj, tryb adresowania i długość całego rozkazu. Dodatkowe słowa zawierają bezpośrednie argumenty lub wyrównania opisane w poprzednim punkcie. Wiele rozkazów może określać rodzaj danych, dopuszczając alternatywny wybór bajtów oraz słów 16- lub 32-bitowych.

Rozkazy przesyłania danych

Rozkazy przesyłania danych tego mikroprocesora są bardzo urozmaicone. Uniwersalna instrukcja MOVE ma dwa argumenty i dla obu może używać dowolnych trybów wyznaczania adresu efektywnego. Znaczy to, między innymi, że dozwolone są przesłania między komórkami pamięci bez pośrednictwa rejestru. W tej grupie rozkazów wyod-

rębniono rozkaz przesyłania adresu, który jako jeden z argumentów musi mieć określony rejestr danych lub rejest adresowy. Dozwolone jest także przesyłanie zawartości całych wybranych grup rejestrów do (z) kolejnych komórek pamięci.

Do tej grupy rozkazów należą także operacje na adresie efektywnym, który może być wprowadzony do rejestru lub zapamiętany na stosie. Rozkazy takie są bardzo przydatne przy tworzeniu kodu z języków wysokiego poziomu w czasie ich tłumaczenia. Rozkazem wygodnym przy organizowaniu wywołań procedur i dynamicznym rezerwowaniu pola na zmienne lokalne przechowywane na stosie jest rozkaz LINK. Jego działanie polega na zapisaniu aktualnej zawartości wskaźnika stosu do rejestru adresowego i zmniejszeniu tego wskaźnika o wartość zadaną w rozkazie. Pole pamięci na stosie wskazywane przez początkową zawartość SP (zmienioną rozkazem LINK) jest właśnie obszarem zarezerwowanym dynamicznie. Dostęp do poszczególnych elementów tego pola można realizować przez odpowiednie tryby adresowania pośredniego przez rejestr. Rozkazem o działaniu odwrotnym jest UNLNK, który odtwarza poprzednią zawartość rejestru adresowego i zwiększa wartość SP zwalniając zarezerwowane pole robocze. Wymienione rozkazy używane przed wywołaniem (LINK) i po wywołaniu procedury (UNLNK) istotnie usprawniają mechanizm uruchamiania procedur rezerwujących dynamicznie pamięć na czas swego działania. Obsługa zmiennych lokalnych procedur w językach wysokiego poziomu (np. PASCAL, ADA), w innych procesorach, nie mających takich rozkazów, jest źródłem znacznego narzutu w ilości generowanego kodu i w czasie jego wykonywania — zarówno w samym wywołaniu procedury, jak i w czasie dostępu do zmiennych lokalnych.

W grupie przesłań międzyrejestrowych znajdują się także rozkazy wymiany poszczególnych połówek rejestrów.

Rozkazy operacji arytmetycznych i logicznych

Podobnie jak inne omawiane mikroprocesory 16-bitowe oraz procesory innych komputerów, mikroprocesor MOTOROLA 68000 może dodawać, odejmować, mnożyć, dzielić i porównywać dwie liczby określone w argumentach rozkazów, a także — zerować, testować, negować i rozszerzać znak przy użyciu rozkazów jednoargumentowych.

Rozkazy dodawania działają na bajtach oraz na słowach 16- i 32-bitowych. Wyodrębnione są rozkazy działań na adresach, które tym różnią się od zwykłych operacji, że nie ustawiają bitów stanu. Operacje dodawania i odejmowania można wykonywać z uwzględnieniem bitu przeniesienia. Oba argumenty tych rozkazów mogą być pobierane ze stosu lub z rejestrów, co jest cechą usprawniającą kod generowany przez translator języka, dla którego standardowa precyzja może nie być wystarczająca. Lista rozkazów zawiera operacje działania na krótkich argumentach bezpośrednich oraz na liczbach przedstawionych na kodzie BCD.

Mnożenie argumentów 16-bitowych generuje wynik 32-bitowy. Jeden z argumentów i wynik są umieszczane w rejestrze. Dzielenie jest zrealizowane analogicznie do mnożenia. Iloraz i resztę dzielenia 32-bitowej dzielnej i 16-bitowego dzielnika zapisuje się na 16 bitach. Oba działania wykonuje się z uwzględnieniem i bez uwzględnienia znaku, co umożliwia łatwe mnożenie i dzielenie z wielokrotną precyzją.

W grupie operacji logicznych występują standardowe rozkazy sumy, różnicy symetrycznej, iloczynu i zaprzeczenia logicznego. Wynik każdej operacji można otrzymywać w rejestrze danych oraz bezpośrednio w pamięci, a argumenty mogą być różnej długości .Wyodrębnione są rozkazy testowania wartości argumentu oraz porównywania dwóch liczb.

Operacje na bitach są reprezentowane przez rozkazy testowania, zerowania, ustawiania i wymiany bitu. Wynik badania jest określony na pozycji **Z** rejestru stanu, a przy ustawieniu, zerowaniu i wymianie — początkowa wartość bitu będącego argumentem jest zapamiętywana na tej pozycji słowa stanu.

Rozkazy przesunięć zawierają przesunięcia arytmetyczne i logiczne oraz cykliczne. Licznik przesunięć jest rejestrem określonym w argumencie rozkazu.

Rozkazy skoków

Rozkazy skoków mikroprocesora MOTOROLA 68000 zawierają: skoki zwykłe, wywołania procedur, skoki warunkowe i skoki warunkowe z badaniem i modyfikacją rejestru. Na liście rozkazów nie ma natomiast warunkowych wywołań procedur. Zbiór warunków dla skoków składa się z 16 elementów. Rozkazy warunkowe używają argumentu 8- lub 16-bitowego opisującego adres względem licznika rozkazów **PC**. Skok zwykły i wywołanie procedury mogą mieć argument względny 8- lub 16-bitowy oraz bezwzględny (pełny).

Skoki warunkowe z modyfikacją rejestru są używane do organizowania iteracji. Ich wykonanie sprowadza się do badania zawartości wskazanego rejestru i wykonania skoku, gdy warunek nie jest spełniony lub testowany rejestr zawiera liczbę —1. W przeciwnym przypadku (warunek spełniony) zawartość rejestru jest zwiększana o 1 i program przechodzi do wykonania następnego rozkazu.

Rozkazy kontroli systemu

Do tej grupy rozkazów zalicza się rozkazy zerowania urządzeń zewnętrznych (RESET), powrotu z przerwania, zatrzymania programu, wpisania nowej wartości do rejestru stanu lub wykonania operacji logicznej na rejestrze stanu procesora, a także przesłania zawartości rejestru stanu pod adres efektywny określony w rozkazie. Wszystkie wymienione rozkazy są wykonywane w trybie systemowym i zgłaszają przerwanie wewnętrzne przy próbie wykonania w trybie normalnym.

Oddzielną grupę rozkazów stanowią przerwania wewnętrzne. Jest wśród nich przerwanie bezwarunkowe, a także przerwanie uaktywniające się przy spełnieniu pewnych warunków. Warunkiem dla rozkazu TRAPV jest ustawienie bitu przepełnienia **V**. Rozkaz CHK jest bardziej wyrafinowany, gdyż porównuje dolne słowo wskazanego rejestru danych z liczbą 16-bitową, określoną przez adres efektywny. Przerwanie wewnętrzne jest skuteczne, gdy zawartość testowanego rejestru jest mniejsza od zera lub większa od liczby określonej adresem efektywnym. Rozkaz CHK umożliwia łatwe badanie, czy indeksy tablic mieszczą się w dopuszczalnych granicach.

PRZERWANIA PROGRAMU

Przyczyną przerwania mogą być następujące zdarzenia wewnętrzne:

- wykonanie rozkazu wewnętrznego przerwania bezwarunkowego
- wykonanie rozkazu TRAPV przy ustawieniu bitu **V** w rejestrze stanu procesora
- wykonanie rozkazu przerwania wewnętrznego CHK z zawartością rejestru spoza badanego zakresu
- próba dzielenia przez zero
- błąd adresu
- próba wykonania nielegalnego rozkazu
- wykonanie niezrealizowanego rozkazu
- wykonanie rozkazu zarezerwowanego dla trybu systemowego
- wykonanie rozkazu normalnego przy ustawieniu bitu **T** w rejestrze stanu procesora.

Zewnętrzną przyczyną przerwania może być: zewnętrzne zgłoszenie, błąd magistrali, zerowanie.

Obsługa przerwania przebiega w następujących fazach:
- przechowanie zawartości rejestru stanu w nieadresowalnych rejestrach wewnętrznych
- ustawienie bitu stanu **S** i skasowanie bitu **T**
- w przypadku przerwania zgłoszeniem wewnętrznym, wpisanie priorytetu na bity maski przerwań w rejestrze stanu w celu uchronienia systemu przed przerwaniem o tym samym lub mniejszym priorytecie
- pomnożenie numeru przerwania przez 4, w celu uzyskania adresu wektora przerwania
- przesłanie bieżącej zawartości licznika rozkazów i przechowanego wcześniej rejestru stanu na stos systemowy
- wpisanie wektora przerwania do licznika rozkazów i przejście do wykonania procedury obsługi.

Najwyższy priorytet przerwania jest używany przez niemaskowalne przerwania typu „zanik napięcia".

JAN DAWIDOWSKI
Akademia Ekonomiczna
Poznań

Komputer osobisty ZX81

Czy komputer osobisty jest w Polsce nadal pojęciem abstrakcyjnym? Na pewno nie, choć jeszcze nie dawno Jacek Żebrowski [7] traktował je w kategoriach nieosiągalnych w Polsce nowości informatycznych. Z przedstawionych przez tego autora różnych sposobów dorobienia się w kraju komputerów osobistych znalazła już realizację ostatnia z dróg — poprzez przedsiębiorstwa polonijno-zagraniczne. Firma AMEPROD z Poznania rozpoczęła pod koniec 1982 roku dostawy komputerów osobistych typu ZX81 na rynek polski.

Czy era nowej, taniej informatyki już się rozpoczęła? Jeżeli nawet nie, to komputer osobisty ZX81 jest z pewnością jaskółką tej ery.

KONSTRUKCJA KOMPUTERA ZX81

Komputer osobisty ZX81, produkt znanej firmy brytyjskiej SINCLAIR, należy do grupy systemów mikrokomputerowych zbudowanych na „jednej płycie drukowanej". Stanowi on wersję rozwojową komputera osobistego ZX80, od którego różni się — oprócz znacznie rozszerzonych możliwości — zmniejszeniem liczby układów scalonych z 21 do... 4 sztuk [6]. W swej wersji podstawowej składa się on z następujących modułów:

— mikroprocesora Z80A
— pamięci stałej (ROM) o pojemności 8 KB
— pamięci roboczej (RAM) o pojemności 1 KB
— układu sterowania
— układów pomocniczych (np. modulatora UHF lub stabilizatora napięcia +5 V).

Wszystkie te układy zmontowane są na płytce drukowanej o wymiarach 100 × 160 mm (!).

Podstawowym urządzeniem wejściowym tego systemu jest miniaturowa klawiatura alfanumeryczna o wymiarach 65 × 160 mm, zawierająca 40 klawiszy, natomiast podstawowe wyjście stanowi standardowy odbiornik TV (czarno-biały), umożliwiający wyświetlenie 24 wierszy po 32 znaki w wierszu.

W wersji standardowej komputer ZX81 wyposażony jest w zewnętrzny blok pamięci roboczej o pojemności 16 KB. Biorąc ponadto pod uwagę elementarne możliwości graficzne (22 znaki graficzne lub 64 × 44 punkty świetlne) monitora ekranowego oraz możliwość przyłączenia popularnego magnetofonu kasetowego jako pamięci zewnętrznej — otrzymamy typowe charakterystyki komputera osobistego [1], wprawdzie bez możliwości przyłączenia minidysków elastycznych, lecz za wielokrotnie niższą cenę.

Opisaną konfigurację można rozszerzyć, a mianowicie:

— zwiększyć pamięć operacyjną do 48 KB
— przyłączyć drukarkę termiczną 32-znakową (szerokość papieru — 100 mm) z możliwością wyprowadzenia rysunków (256 punktów w wierszu)
— przyłączyć drukarkę mozaikową EPSON MX-80
— zastosować złącze RS 232 C (V-24) itd.

Dostawca kompletuje komputery w dwa podstawowe zestawy:

• **ZX81—01**, składający się z następujących modułów:

— podstawowego ZX81
— pamięci ZX 16 K RAM
— drukarki termicznej ZX Printer

oraz wyposażenia dodatkowego (zasilacz, oprogramowanie podstawowe, dwie kasety, dokumentacja techniczno-ruchowa, dokumentacja oprogramowania, dodatkowa rolka papieru do drukarki).
Cena 148 tys. zł.

• **ZX81—02**, składający się z następujących modułów:

— podstawowego ZX81
— pamięci ZX 64 K RAM (48 KB pamięci użytkowej)
— drukarki termicznej ZX Printer
— wyposażenia dodatkowego (zasilacz, oprogramowanie podstawowe, dwie kasety, dokumentacja techniczno-ruchowa, dokumentacja oprogramowania, dodatkowa rolka papieru do drukarki).
Cena 195 tys. zł.

Ponadto dostawca kompletuje powyższe zestawy bez drukarki termicznej (ceny — odpowiednio — 92 tys. zł i 145 tys. zł), a nawet bez modułu pamięci (cena — 52 tys. zł). Jedyny materiał eksploatacyjny — papier do drukarki termicznej — dostępny jest w cenie 2100 zł za rolkę (ok. 20 mb).

JĘZYK ZX81 BASIC

Podstawowym językiem komputera ZX81 jest wersja szeroko znanego i łatwego w opanowaniu języka BASIC [2]. Umożliwia ona prowadzenie prostych obliczeń, a także realizację bardziej złożonych algorytmów w warunkach wielokrotnego wykonywania operacji. W obu przypadkach komputer wykonuje operacje na dwóch podstawowych typach danych:

• zmiennych lub elementach tablic liczbowych (również wielowymiarowych), mogących reprezentować liczby z przedziału $(10^{-\epsilon 8}, 10^{+38})$ z dokładnością ponad 9 cyfr dziesiętnych (mantysa — 4 bajty, wykładnik — 1 bajt).
• zmiennych lub elementach tablic tekstowych (również wielowymiarowych), reprezentujących wartości znakowe (stałej lub zmiennej długości).

Na liczbach dozwolone jest stosowanie:

— operacji podstawiania (LET... = ...)
— czterech działań arytmetycznych (+, —, ×, /)
— funkcji standardowych: * * (postępowanie), SIN, COS, TAN, ARCSIN, ARCCOS, ARCTAN, INT, SGN, ABS, SQR, LN, EXP, RND (wszystkie z dokładnością do 8 cyfr)
— dowolnej liczby nawiasów
— sześciu operacji porównywania wielkości liczbowych
— wyrażeń logicznych jako wielkości liczbowych (o wartościach odpowiednio 0 i 1)
— skomplikowanych wyrażeń logicznych — za pomocą operacji AND, OR i NOT oraz ich rozszerzonej interpretacji
— umożliwiających łączenie wyrażeń arytmetycznych i logicznych
— konwersji wielkości liczbowych na ich postać znakową (funkcja STR$)

— konwersji wielkości liczbowych na pojedyncze znaki o odpowiednich kodach (funkcja CHR$).

Na tekstach dozwolone jest stosowanie:
— operacji podstawiania (LET ... = ...)
— operacji konkatenacji (+)
— operacji wybrania podciągu z danego ciągu (po obu stronach operacji podstawiania)
— funkcji określania długości tekstu i kodu danego znaku (funkcje LEN i CODE)
— operacji obliczania wartości wyrażenia arytmetycznego reprezentowanego przez dany tekst (funkcja VAL)
— rozszerzonej interpretacji wyrażeń logicznych AND i OR, umożliwiających łączenie wyrażeń tekstowych i logicznych.

Formułowanie algorytmów ułatwiają instrukcje:
— skoku GOTO...
— warunkowa IF...THEN... o rozszerzonych możliwościach (po słowie THEN może występować dowolny operator języka BASIC)
— pętli FOR...TO...STEP... i NEXT (maksymalnie 26 poziomów zagłębienia)
— podprogramu GOSUB i RETURN
— wprowadzania danych z klawiatury INPUT...
— wyprowadzania wyników PRINT i LPRINT, z opcjami AT, TAB, ", ", ", "
— korzystanie z ekranu TV — CLS i SCROLL oraz wyświetlanie na ekranie rysunków — PLOT i UNPLOT
— definicji rozmiaru tablic DIM
— pomocnicze PAUSE i STOP
— komentarza REM.

Wprowadzanie, poprawianie i wykonywanie algorytmów ułatwiają natomiast dyrektywy:
— zerowania pamięci komputera NEW
— zerowania tylko obszaru zmiennych CLEAR
— uruchomienia programu RUN
— kontynuowania wykonywania przerwanego programu CONT
— ładowania na taśmę magnetyczną i ładowania z taśmy magnetycznej całej pamięci komputera — SAVE i LOAD
— listowania programu i kopiowania zawartości ekranu na drukarce termicznej — LIST, LLIST i COPY
— zmiany trybu pracy komputera SLOW i FAST.

Dostęp do pojedynczych bajtów pamięci realizują — instrukcja POKE i funkcja PEEK.

Język BASIC umożliwia również wykorzystywanie za pomocą funkcji USR wstawek w języku wewnętrznym mikroprocesora Z80A.

Wszystkie te możliwości komputera ZX81, a więc: interpreter języka BASIC, system operacyjny i edytor tekstów programów, są zapisane w 8 KB pamięci stałej (ROM).

MOŻLIWOŚCI FORMUŁOWANIA I WYKONYWANIA ALGORYTMÓW

W celu rozpoznania możliwości obliczeniowych komputera ZX81, opracowano w języku BASIC dwa algorytmy typowo numeryczne, dotyczące obliczania wartości własnych dowolnej macierzy rzeczywistej metodą „QR" i odwracania macierzy metodą Gaussa-Jourdana. Oba algorytmy umożliwiają na komputerze ZX81—01 (tzn. z pamięcią roboczą 16 KB) rozwiązywanie wymienionych zagadnień dla macierzy stopnia 50 (!). Orientacyjne czasy wykonywania tych algorytmów dla różnych stopni macierzy przedstawia tabela.

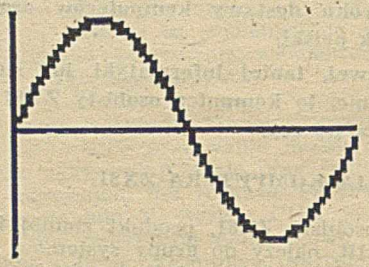
Jako przykład algorytmu nienumerycznego opracowano tzw. disasembler języka wewnętrznego mikroprocesora Z80, odtwarzającego postać mnemoniczną rozkazów na podstawie kodu wynikowego programu. Orientacyjny czas wykonania programu, niezbędny dla wypełnienia 22 wierszy ekranu, wynosił ok. 12 sekund, zaś wyprowadzenie wyników na drukarce termicznej — ok. 9 sekund.

Orientacyjne czasy wykonania operacji obliczania wartości własnych macierzy i odwracania macierzy na komputerze osobistym ZX81-01

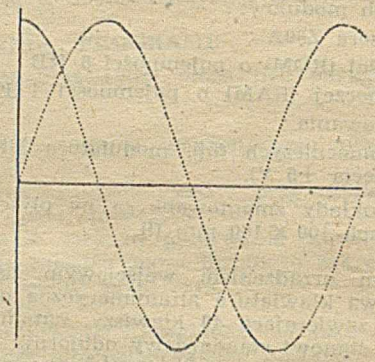
Stopień macierzy	Czas wykonania operacji	
	obliczanie wartości własnych	odwracanie macierzy
5	50 sekund	6 sekund
10	4 minuty	30 sekund
20	25 minut	4 minuty
30	1 godzina	13 minut
40	2 godziny	30 minut
50	3,5 godziny	1 godzina
95*	15 godzin*	5 godzin*

* Maksymalny stopień macierzy możliwy do obliczeń na komputerze ZX81-02, dane szacunkowe

Standardowe możliwości wyprowadzania wyników w postaci wykresów na ekran TV ilustruje rysunek 1, natomiast rysunek 2 — maksymalną rozdzielczość drukarki termicznej (na przykładzie niestandardowo wyprowadzanego rysunku).



Rys. 1. Standardowa postać wyników programu na ekranie TV



Rys. 2. Maksymalna rozdzielczość drukarki termicznej

Praktyczne próby potwierdziły także możliwość zapisywania na jednej stronie kasety C-60 ponad 60 KB informacji (przy użyciu popularnego magnetofonu MK 122). Pozwala to na kompletowanie bibliotek kilkunastu sporych programów na jednej stronie kasety.

Wprowadzenie, poprawienie i uruchomienie powyższych algorytmów pochłonęło odpowiednio 3 godz., 1 godz. i 6 godz.

MOŻLIWOŚCI ZASTOSOWAŃ

ZX81 jest minikomputerem, który dzięki swoim walorom nadaje się szczególnie do takich tradycyjnych dziedzin zastosowań, jak:

- wszelkie doraźne obliczenia
- obliczenia naukowe, a w szczególności — eksperymenty numeryczne z modelami, gry kierownicze itp. [5]

- obliczenia inżynierskie
- emulacja inteligentnej stacji końcowej dużego systemu komputerowego
- wspomaganie procesu dydaktyki w szkolnictwie średnim i wyższym, nie tylko z zakresu informatyki.

DOŚWIADCZENIA EKSPLOATACYJNE

Miesięczna eksploatacja komputera osobistego ZX81-01 pozwala sformułować szereg uwag dotyczących zarówno jego niewątpliwych zalet, jak i wad. I tak do zalet należy zaliczyć:

— czytelność i łatwość opanowania wygodnej klawiatury (do pięciu funkcji na pojedynczym klawiszu)
— krótki czas reakcji systemu
— duża elastyczność języka BASIC
— czytelny obraz na ekranie TV
— możliwość rozwiązywania poważnych problemów numerycznych w dopuszczalnym czasie (i to w warunkach domowych!)
— łatwość rozbudowy konfiguracji.

Interesująca jest również zapowiadana możliwość rozbudowy konfiguracji systemu na drodze wymiany modułów (np. pamięciowych) i dostawy wersji rozwojowej tego komputera — ZX-Spectrum.

Do głównych wad komputera zaliczyć natomiast należy:
— niedostosowanie końcówek kabli połączeniowych do krajowych magnetofonów i odbiorników TV
— niestabilność połączeń modułów dodatkowych z modułem ZX81, spowodowaną brakiem konstrukcji wspornej, poza złączem
— brak wyłącznika zasilania, powodująca konieczność wyłączania komputera przez wyjmowanie wtyczki z gniazda sieci elektrycznej
— niekompletność dokumentacji oprogramowania, a w szczególności — brak tekstów programów, praktycznie uniemożliwia odtworzenie zawartości kaset w przypadku ich zniszczenia.

Innego podejścia wymaga natomiast sprawa szkoleń i serwisu oprogramowania, która dla tego typu systemów komputerowych winna znaleźć swoje własne ramy organizacyjne (np. Klub Użytkowników).

•

Czy jest to pierwsza prezentacja komputera osobistego ZX81 na rynku polskim? Na pewno nie. Po raz pierwszy był on demonstrowany na 54 Międzynarodowych Targach Poznańskich w 1982 roku (przez firmę polonijną AMEPROD), gdzie został wyróżniony medalem Oddziału Zachodniego Polsko-Polonijnej Izby Przemysłowo-Handlowej „INTER-POLCOM" w Poznaniu oraz Klubu Producentów i Handlowców Polonijno-Zagranicznych SALPOL. Niestety, nie znalazło to odzwierciedlenia w INFORMATYCE, w skądinąd ciekawej relacji z tych targów [8], gdzie wiele miejsca poświęcono innym firmom, w tym polonijnym.

Komputer osobisty ZX81 wszedł już na rynek polski, wypełniając lukę pomiędzy tanimi „Bolkami" a szalenie drogimi i praktycznie nieosiągalnymi systemami mini- i superkomputerowymi, dostarczanymi przez krajowy przemysł informatyczny.

LITERATURA

[1] Dokumentacja użytkowa — mikrokomputer ZX-81, podłączenie i uruchomienie
[2] Dokumentacja użytkowa — ZX-81 BASIC, podstawy programowania. Zakład Projektowania Systemów Sterowania, Katowice
[3] Dokumentacja oprogramowania podstawowego. Zakład Projektowania Systemów Sterowania, Katowice
[4] Ellison D., Herschdorfer I., Wilson J. T.: Interactive simulation on a microcomputer. Simulation, 1982, 38, nr 5, 161—175 (Ekspres Informacja, Wyczislit. Techn., 1982, nr 2, 24—28, Moskwa)
[5] Informacja ofertowa firmy AMEPROD
[6] Vier ICs machen einen BASIC-Rechner. Chip, 1981, nr 9, 34—36 (Referat. Żurnal, Awtomat. i wyczislit. techn. 1982, 1B258, Moskwa)
[7] Żebrowski J.: Komputery osobiste. INFORMATYKA, nr 6/82, 6—9
[8] Żebrowski J.: 54 MTP, Inwazja mikroprocesorów, INFORMATYKA, nr 7/82, 28—31.

„CENTROSTAL"

Przedsiębiorstwo Państwowe Obrotu Wyrobami Hutniczymi

ul. Waryńskiego 7, 78-400 Szczecinek

zgłasza do upłynnienia niżej wyszczególnione nowe maszyny produkcji NRD (okres eksploatacji 3 miesiące):

- dziurkarka kart „Soemtron" S-415 — szt. 1,
- sprawdzarka kart perforowanych „Soemtron" S-425 — szt. 1,
- kalkulatory „Iskra" — szt. 2,
- waga elektryczna typu ZUP-1 — szt. 2.

Zamówienia i ewentualną korespondencję prosimy nadsyłać na wyżej wymieniony akres, telefon 425-11 wew. 65. Dział Administracji.

EO/615/K/83

POMORSKI OKRĘGOWY ZAKŁAD GAZOWNICTWA w Gdańsku

Zakład Gazowniczy Bydgoszcz

zakupi:

pamięć dyskową MERA 9425

Informacji udziela i oferty przyjmuje:

Zakład Gazowniczy Bydgoszcz
ul. Jagiellońska 42
85-097 Bydgoszcz
tel. 22-00-81 wew. 320

EO/659/K/83

MARIAN FURMAN, KONRAD KARPIŃSKI,
ZBIGNIEW NOWICKI, JERZY SOLAK

Instytut Informacji Naukowej,
Technicznej i Ekonomicznej
Warszawa

Rozszerzenie pakietu informacyjno-wyszukiwawczego SIM-400M

Niniejsze opracowanie stanowi skrótowy opis rozszerzeń i zmian wprowadzonych do pakietu informacyjno-wyszukiwawczego SIM-400M od momentu opublikowania w INFORMATYCE pierwszej informacji na jego temat [1]. Podobnie jak podstawowa wersja pakietu, rozszerzenia te zostały wykonane przez zespół Zakładu Struktur Sieci Informacyjnej Instytutu INTE [1]).

SIM-400M jest pakietem programów umożliwiających gromadzenie, przetwarzanie i wyszukiwanie informacji za pomocą minikomputera MERA-400. W swej pierwotnej wersji składa się on z ośmiu niezależnych modułów programowych, realizujących podstawowe operacje na zbiorach danych. Są to:

EDYTOR — moduł umożliwiający zakładanie zbiorów danych, wprowadzanie do nich nowych danych i kontekstowe wymiany w ramach istniejących w zbiorze dokumentów

DRUKUJ — program umożliwiający wydruk wybranych fragmentów wskazanych dokumentów zbioru

SCALAJ — program pozwalający łączyć dwa zbiory, zawierające dokumenty o podobnej strukturze, w jeden zbiór

SORTUJ — moduł pozwalający posortować zbiór dokumentów według zawartości wskazanych pól

SŁOWNIK — program sporządzający alfabetyczny wykaz wszystkich słów użytych w wybranych polach wskazanych dokumentów

INDEKS — moduł sporządzający alfabetyczny wykaz wszystkich deskryptorów znalezionych w polu deskryptorowym, wraz z listą numerów odpowiednich dokumentów

SZUKAJ — program pozwalający wyszukiwać dokumenty zbioru według warunków narzuconych dla dowolnych pól dokumentu

OBLICZ — moduł umożliwiający wykonywanie operacji arytmetycznych na polach liczbowych dokumentów.

Wymienione moduły przystosowane są do pracy w trybie wsadowym, a ich wykonanie jest sterowane za pomocą odpowiednich zestawów dyrektyw.

Każdy zbiór przystosowany do przetwarzania przez programy pakietu składa się z dokumentów jednego rodzaju. Dokumenty umieszczone we wspólnym zbiorze są podzielone na pola opatrzone identyfikatorami. Każde pole jest typu znakowego lub liczbowego, co oznacza, że może zawierać ciąg znaków lub pojedynczą liczbę. Długość dokumentu dostosowana jest do jego zawartości, dzięki czemu uzyskuje się bardzo dobre upakowanie danych w pamięci zewnętrznej.

[1]) Pełny opis zmian zawiera dokumentacja uzupełniająca pakietu [5, 6, 7], którą osoby zainteresowane mogą uzyskać u producenta

Dokładne informacje, dotyczące wyżej wymienionych modułów i organizacji zbiorów danych można znaleźć w opracowaniach [2, 3, 4].

Kontynuacja prac nad pakietem SIM-400M była podyktowana dużym zainteresowaniem użytkowników, których uwagi w decydujący sposób wpłynęły na kierunki rozwoju pakietu. Zgłaszano zapotrzebowanie na nowe moduły programowe, które m.in. pozwalałyby w trybie konwersacyjnym prowadzić wyszukiwanie i przygotowywać nowe dokumenty. Jednocześnie, ze względu na znacznie większą — niż się spodziewano pierwotnie — różnorodność zastosowań pakietu (jest on stosowany do przetwarzania zarówno typowych zbiorów bibliotecznych, jak i zbiorów zawierających dane liczbowe, na których wykonuje się złożone działania rachunkowe), konieczne stało się wprowadzenie do już istniejących modułów wsadowych takich zmian, które sprzyjałyby pełnemu wykorzystaniu możliwości pakietu dla szerokiego zakresu zastosowań. Zmiany te dotyczyły przede wszystkim modułów SZUKAJ i OBLICZ.

Przy okazji tych prac dokonano szeregu rozszerzeń i ulepszeń, których celem było ułatwienie korzystania z pakietu i uogólnienie stosowanych w nim mechanizmów. W rezultacie istotnym zmianom uległy moduły: SZUKAJ, OBLICZ i DRUKUJ. Powstały także dwa nowe moduły przystosowane do działania w trybie konwersacyjnym. Są to:

DIALOG — program pozwalający prowadzić wyszukiwanie w sposób konwersacyjny

POMOCNIK — program wspomagający przygotowanie dokumentów wejściowych.

Na podkreślenie zasługuje fakt, że nie zostały naruszone dotychczasowe reguły organizacji zbiorów, dzięki czemu zbiory dokumentów przygotowane przy użyciu podstawowej wersji pakietu mogą być przetwarzane także przez moduły nowej wersji.

WYSZUKIWANIE

W podstawowej wersji modułu SZUKAJ, kryterium wyszukiwawczym było wystąpienie wewnątrz wskazanego pola dokumentu słowa, rozpoczynającego się od podanego wzorca znakowego. Za pomocą operatorów logicznych i nawiasów można było z kilku kryteriów tego typu utworzyć kryterium złożone, pozwalające uzależnić wybór dokumentu od występowania różnych wzorców znakowych w różnych polach dokumentu.

W obecnej wersji wzbogacono repertuar relacji pomiędzy zawartością pola i wzorcem, jakie mogą być badane w trakcie wyszukiwania. Obok istniejącej dotychczas relacji występowania słów (=?), możliwe jest badanie relacji równości (=), większości (>), mniejszości (<) oraz relacji złożonych z wyżej wymienionych relacji prostych (np. >=), przy czym wszystkie te relacje dotyczą pełnej zawartości pola.

Nowo wprowadzone relacje są interpretowane w różny sposób, zależnie od typu badanego pola i wzorca. Jeśli pole

i wzorzec są typu liczbowego, relacja ma zwyły sens liczbowy. Jeśli wzorzec jest typu znakowego (wzorzec taki jest ciągiem znaków, ograniczonym znakami gwiazdki), a pole jest znakowe lub liczbowe, to relację traktuje się jako leksykograficzną. Spełnienie takiej relacji bada się przez porównanie ze sobą kolejnych znaków pola i wzorca. Podobnie jak w poprzedniej wersji, złożone kryterium wyszukiwawcze można sformułować, łącząc relacje za pomocą nawiasów oraz operatorów logicznych koniukcji (&), alternatywy (#) i negacji (/\).

Załóżmy, że dostępny jest zbiór STYPENDYŚCI, którego dokumenty zawierają następujące pola wypełnione danymi osób pobierających stypendia:

NAZW — nazwisko

IMIE — imię

ADR — adres zamieszkania

SRDO — średni dochód na członka rodziny

SOCE — średnia z ocen uzyskanych w czasie ostatniej sesji egzaminacyjnej

STYP — wysokości stypendium,

przy czym trzy pierwsze pola są polami znakowymi, a trzy pozostałe — numerycznymi. Wszystkich stypendystów zamieszkałych w stolicy, którzy uzyskali średnią z ocen nie mniejszą niż 4, można wyszukać zadając pytanie:

```
/SZUKAJ/ ADR =? *WARSZAWA* & SOCE>= 4
```

W pytaniu tym w odniesieniu do pola ADR użyto relacji występowania, a nie relacji równości — ze względu na to, że adres może zawierać, oprócz nazwy miasta, nazwę ulicy, nr domu itp.

OBLICZENIA

W najnowszej wersji pakietu możliwości modułu OBLICZ, służącego do wykonywania operacji arytmetycznych w obrębie zbioru dokumentów, uległy znacznemu rozszerzeniu w stosunku do wersji podstawowej. Stało się to dzięki umożliwieniu uzależnienia wykonania działań od spełnienia relacji arytmetycznych dotyczących pól liczbowych dokumentu lub pomocniczych zmiennych liczbowych.

Wykonaniem obliczeń przez moduł sterują trzy dyrektywy. Są to:

/WYBIERZ/ — dyrektywa wskazuje te dokumenty, które mają być uwzględnione w trakcie przetwarzania

/DEKLARUJ/ — dyrektywa służy do deklarowania pomocniczych zmiennych liczbowych i nadawania im wartości początkowych

/OBLICZ/ — dyrektywa zawiera szczegółowy opis operacji, które mają być wykonywane dla każdego z wybranych dokumentów.

Opis działań zawarty w dyrektywie /OBLICZ/ ma postać programu, w którym można wyróżnić instrukcje podstawienia, warunku i skoku.

Instrukcja podstawienia umożliwia nadanie nowej wartości zmiennej lub polu liczbowemu dokumentu. Wartość ta jest określona za pomocą wyrażenia arytmetycznego zbudowanego z identyfikatorów pól liczbowych dokumentu, identyfikatorów zmiennych pomocniczych, stałych liczbowych i operatorów czterech podstawowych działań arytmetycznych oraz nawiasów. Przykładowa instrukcja podstawienia, ustalająca wysokość stypendium STYP, zależnie od wartości zmiennych pomocniczych określających: stawkę podstawową STPO i dodatek DOD, może mieć postać:

```
STYP := 0.8 * STPO + DOD
```

Instrukcja skoku pozwala w dowolny sposób ustalić miejsce w programie od którego kontynuowane będzie wykonanie programu po wykonaniu tej instrukcji. Ma ona postać:

```
SKOCZ etykieta
```

przy czym etykieta identyfikuje dokładnie jedno miejsce w programie.

Instrukcja warunku ma postać relacji pomiędzy polem liczbowym lub zmienną pomocniczą, a wyrażeniem arytmetycznym zbudowanym w ten sam sposób jak w instrukcji podstawienia. Typ relacji wskazuje się tak samo jak przy wyszukiwaniu (=, >, < lub kombinacja tych symboli). W przypadku, gdy w momencie wykonania instrukcji warunku podana relacja nie jest spełniona, omijana jest najbliższa instrukcja podstawienia lub skoku występująca za tą instrukcją. W przeciwnym razie kontynuowane jest sekwencyjne wykonywanie programu.

Można udowodnić, że te proste środki programowe pozwalają uzależnić wykonanie spójnego fragmentu programu od spełnienia dowolnego warunku, skonstruowanego z prostych relacji arytmetycznych połączonych operatorami logicznymi alternatywy koniukcji i negacji.

Poniżej przedstawiono przykładowy program, który powoduje zwiększenie wysokości stypendium tych studentów, którzy osiągnęli dobre wyniki w nauce:

```
/OBLICZ/ (SOCE < 4 ,
           SKOCZ E,
         SOCE < 4.5
           STYP : = STYP + DOD1,
         SOCE>= 4.5
           STYP := STYP + DOD2,
         LIC := LIC + 1,
       E :)
```

W programie tym, oprócz identyfikatorów liczbowych pól dokumentów STYP i SOCE, występują zmienne pomocnicze: DOD1, DOD2 i LIC. Ich początkowe wartości muszą być ustalone za pomocą dyrektywy DEKLARUJ. DOD1 określa wysokość dodatku do stypendium w przypadku, gdy średnia z ocen jest niższa od 4,5 lecz nie jest mniejsza od 4. Jeśli średnia z ocen osiąga co najmniej 4,5, to dodatek ma wartość DOD2. Jeżeli początkową wartością zmiennej LIC jest 0, to po przetworzeniu całego zbioru dokumentów zmienna ta podaje liczbę wszystkich stypendystów, którzy otrzymują dodatek do stypendium z tytułu dobrych wyników w nauce.

Powyższy przykład ilustruje możliwość wykonywania zarówno obliczeń lokalnych (w ramach każdego przetworzanego dokumentu), jak i obliczeń globalnych (obejmujących cały zbiór dokumentów).

WYDRUK TABELARYCZNY

Moduł DRUKUJ w swej dotychczasowej postaci umożliwia wydruk wybranych pól dokumentu w oddzielnych wierszach. Taki sposób wyprowadzania danych jest uzasadniony wtedy, gdy poszczególne pola dokumentu przechowują dużą ilość danych. Jeśli jednak pola przeznaczone do wydruku zawierają liczby lub krótkie teksty, wówczas wygodnie jest korzystać z wydruku w postaci tabeli, w której jeden wiersz zawiera dane pochodzące z jednego dokumentu. Taki właśnie sposób drukowania może być zlecony za pomocą nowo wprowadzonej dyrektywy TABELA. Pozwala ona określić szerokości kolumn odpowiadających poszczególnym polom, sposób rozmieszczania tekstu w kolumnie, teksty drukowane pomiędzy kolumnami, a także treść i sposób wydruku nagłówka tabeli.

Listę pól, których zawartość ma być wydrukowana, określa się za pomocą dyrektywy OGRANICZ POLA. Opisy poszczególnych kolumn tabeli łączone są z elementami tej listy na podobnej zasadzie, jak w FORTRANIE kojarzy się opisy pól z deklaracji FORMAT z elementami listy wejścia-wyjścia. Opis kolumny, obok obowiązkowego określenia szerokości kolumny, może zawierać informację o usytuowaniu kropki dziesiętnej, jeśli w kolumnie mają być drukowane liczby i może ustalać treść nagłówka kolumny, jeśli nie ma nim być identyfikator odpowiedniego pola. Jeśli pomiędzy opisami kolumn są umieszczone jakieś teksty, to w czasie wydruku tabeli są one powtarzane w odpowiednim miejscu każdego wiersza.

Przykładowo — przy przetwarzaniu zbioru STYPENDYSCI, podanie w zestawie dyrektyw modułu DRUKUJ następującej pary dyrektyw:

```
/OGRANICZ POLA/ (NAZW, IMIE, SOCE)
(TABELA) (*<15, NAZWISKO>*<15>*<1.2> *)
```

spowoduje, że z każdego dokumentu będą drukowane tylko trzy pola' NAZW, IMIE, SOCE. Kolumny przeznaczone na nazwisko i imię będą miały po 15 znaków szerokości, a średnia z ocen będzie zajmować kolumnę o szerokości 4 znaków, przy czym pierwszą pozycję będzie w tym przypadku zajmować część całkowita liczby, pozycję drugą — kropka, a dwie ostatnie pozycje są przeznaczone na część ułamkową. W nagłówku pierwszej kolumny figurować będzie tekst: NAZWISKO, a dwie pozostałe kolumny będą miały w nagłówku nazwy odpowiadających im pól: IMIE i SOCE. Znak gwiazdki będzie rozpoczynał i kończył każdy wiersz tabeli, a także będzie oddzielał poszczególne kolumny. Przykład wydruku takiej tabeli pokazano na rysunku.

* NAZWISKO	* IMIE	* SOCE *
* BADURA	* ANTONI	* 3.23 *
* BORZYNSKI	* ZENON	* 4.07 *
* CUDNY	* MALGORZATA	* 3.41 *

Przykładowy fragment wydruku tabelarycznego

Za pomocą dodatkowego parametru dyrektywy TABELA można ustalać różne sposoby opatrywania tabeli nagłówkiem: nagłówek tylko na początku tabeli, nagłówek na początku każdej strony lub nagłówek na początku i na końcu każdej strony. Ponadto można wydzielić ostatni wiersz, jeśli dukowany w nim dokument zawiera np. podsumowanie całej tabeli.

WYSZUKIWANIE KONWERSACYJNE

Omówione dotychczas rozszerzenia pakietu SIM-400M miały charakter zmian odniesionych do istniejących modułów wsadowych. Pakiet został także wzbogacony o dwa moduły konwersacyjne, pozwalające użytkownikowi prowadzić dialog z systemem cyfrowym za pośrednictwem monitora ekranowego z klawiaturą lub za pomocą konsoli KSR.

Moduł DIALOG umożliwia wyszukiwanie w dowolnie wybranym zbiorze, znajdującym się w aktualnie dostępnej pamięci dyskowej, pod warunkiem znajomości hasła dostępu do tego zbioru. Pytanie wyszukiwawcze ma postać podobną jak w module SZUKAJ. Wyszukane dokumenty tworzą zbiór wynikowy, który może być przez użytkownika przeglądany drogą wyświetlania dokumentu po dokumencie na ekranie monitora. W trakcie przeglądania użytkownik może zlecić wydruk poszczególnych dokumentów lub ich umieszczenie w nowym zbiorze wynikowym (realizując w ten sposób wyszukiwanie manualne). Jeśli zbiór wynikowy zawiera zbyt wiele dokumentów lub nie zawiera wszystkich dokumentów interesujących użytkownika, może on — zadając kolejne pytanie wyszukiwawcze — zlecić zawężenie tego zbioru lub jego rozszerzenie o nowe dokumenty wyszukane w zbiorze źródłowym.

Poprzedzenie pytania wyszukiwawczego symbolem & oznacza, że jest ono skierowane do aktualnie dostępnego zbioru wynikowego, a poprzedzenie go symbolem # oznacza, że przeszukiwany ma być zbiór źródłowy, lecz wyniki wyszukiwawcze powinny być dołączone do aktualnie dostępnego zbioru wynikowego. Jeśli dostępne zasoby pamięci dyskowej pozwalają na to, przechowywane są dwie ostatnie wersje zbioru wynikowego. Wówczas użytkownik ma możliwość cofnięcia się do wersji poprzedniej.

Każdy ze zbiorów utworzonych w trakcie konwersacji może być wydrukowany w całości lub umieszczony w wydzielonym obszarze w pamięci masowej wraz z żądanym komentarzem.

W trakcie konwersacji użytkownik korzysta z szeregu udogodnień wspomagających współpracę z modułem. Jednym z nich jest możliwość korzystania z informatora przygotowanego przez administratora systemu.

Poniższy przykład ilustruje tworzenie kolejnych wersji zbioru wynikowego dla zbioru źródłowego STYPENDYSCI, poprzez zadawanie pytań rozszerzających i zawężających:

```
? SZUKAJ SOCE > 3.5 & ADR =? *WARSZAWA*
WYSZUKANO 78 DOKUMENTOW
? SZUKAJ & SOCE > 4
WYSZUKANO 40 DOKUMENTOW
? SZUKAJ # SRDO < 2000
WYSZUKANO 45 DOKUMENTOW
? DRUKUJ
```

Wiersze rozpoczynające się od znaku zapytania zawierają dyrektywy użytkownika, a pozostało wiersze są odpowiedziami systemu. W rezultacie przeprowadzenia tego krótkiego fragmentu dialogu utworzono zbiór zawierający wszystkie dokumenty dotyczące stypendystów o dochodzie na członka rodziny mniejszym od 2000 zł lub mieszkających w Warszawie i mających średnią z ocen powyżej 4. Następnie zbiór ten został wydrukowany.

PRZYGOTOWANIE DOKUMENTÓW WEJŚCIOWYCH

Drugim modułem działającym w trybie konwersacyjnym jest POMOCNIK. Zadaniem tego modułu jest ułatwienie przygotowania dokumentów źródłowych, z których ma być utworzony nowy zbiór lub które mają zostać dołączone do zbioru założonego wcześniej. Główną trudnością jest tutaj zachowanie ustalonego formatu danych wejściowych i wprowadzenie pól dokumentu w odpowiedniej kolejności. Obydwie te trudności mogą być złagodzone przez moduł POMOCNIK, pod warunkiem, że dostarczy się mu informacji o budowie dokumentu. Istnieje również możliwość poprawienia pomyłek do momentu zatwierdzenia dokumentu.

* * *

Ważną cechą obecnej wersji pakietu SIM-400M jest możliwość korzystania przez użytkownika z silnego aparatu wyszukiwawczego. Pozwala on zadawać złożone pytania, w których można się odwołać do pełnego tekstu dokumentu. Wymaga to oczywiście przeglądania całego zbioru dokumentów w trakcie poszukiwania odpowiedzi na pytanie.

Zazwyczaj w dokumentach pewnych typów można wyróżnić grupę pól szczególnie często pojawiających się w pytaniach. Czas wyszukiwania odpowiedzi na pytania odwołujące się do tych pól może być skrócony dzięki zastosowaniu indeksów i list odwróconych. Realizacja tego typu metod przyśpieszających wyszukiwanie stanowi jeden z najbliższych celów zespołu pracującego nad pakietem. Innym ważnym kierunkiem prac będzie umożliwienie odpowiedzi na pytania odwołujące się do wielu zbiorów jednocześnie.

Wszelkie informacje dotyczące obecnej wersji pakietu SIM-400M, a także planowanych rozszerzeń można uzyskać w Zakładzie Struktur Sieci Informacyjnej Instytutu INTE.

LITERATURA

[1] Bańkowski J. i in.: Uniwersalny pakiet przetwarzania i wyszukiwania informacji dla MERY-400. INFORMATYKA, 9/1980, str. 15—17

[2] Minikomputerowy system informacyjno-wyszukiwawczy SIM-400M. Prace IINTE, 36, Warszawa, 1981

[3] System informacyjno-wyszukiwawczy SIM-400. Instrukcja użytkownika, (maszynopis), IINTE, Warszawa, 1979

[4] System informacyjno-wyszukiwawczy SIM-400. Dokumentacja uzupełniająca 1/80 (maszynopis), IINTE, Warszawa, 1980

[5] System informacyjno-wyszukiwawczy SIM-400M. Uzupełniająca dokumentaeja operatorska 1/81 (maszynopis), IINTE, Warszawa, 1981

[6] System informacyjno-wyszukiwawczy SIM-400M. Uzupełniająca dokumentacja użytkowa 2/81 (maszynopis), IINTE, Warszawa, 1981

[7] System informacyjno-wyszukiwawczy SIM-400M. Dokumentacja uzupełniająca 1/82 (maszynopis), IINTE, Warszawa 1982.

ZBIGNIEW REMPAŁA
Zakłady Azotowe
Tarnów

ROMAN RYGAŁ
Instytut Elektroenergetyki
Politechnika Częstochowska

Translator języka adresów bezwzględnych dla procesora autonomicznego CAMAC typu 131

Procesor autonomiczny typu 131 [4] umożliwia konstruowanie w systemie CAMAC [3] urządzeń sterujących o czasie reakcji rzędu mikrosekund. Jednakże oferowane przez wytwórcę oprogramowanie, wymagające użycia pamięci operacyjnej o pojemności 2 K słów 24-bitowych jest obecnie nieprzydatne, ponieważ bloki pamięci o takiej pojemności nie są produkowane w kraju. Użytkownik staje wobec konieczności tworzenia własnego oprogramowania w postaci bezpośrednio akceptowanej przez standardowy program ładujący, co z kolei wydłuża proces uruchamiania programów. Alternatywne postępowanie polega na stosowaniu translatorów skrośnych, działających na dużych maszynach cyfrowych.

Translator skrośny powinien:

— dokonywać tłumaczenia programu źródłowego na kod maszynowy

— umożliwiać tworzenie programów bibliotecznych i dołączanie ich do programu wynikowego

— sygnalizować błędy w programie źródłowym

— wyprowadzać opis programu na drukarkę wierszową, a program wynikowy — na taśmę papierową.

Translator o powyższych właściwościach uruchomiono na maszynie cyfrowej ODRA 1325, w ramach prac nad systemem ATA [1].

CHARAKTERYSTYKA JĘZYKA ŹRÓDŁOWEGO

Język zrozumiały dla translatora jest w zasadzie równoważny językowi CAMASS [2]. Dokonano w nim kilku zmian, które ułatwiają programowanie oraz umożliwiają wzrokowe wykrycie błędnie napisanych instrukcji. Zmiany te nie powinny stanowić żadnego utrudnienia dla użytkownika znającego język CAMASS. Program źródłowy składa się z ciągu wierszy zakończonego wierszem końca programu. Każdy z tych wierszy zawiera trzy pola wzajemnie oddzielone pojedynczym odstępem. Kolejność pól jest następująca: pole rozkazu, pole warunku, pole argumentu.

Pole rozkazu jest przeznaczone na trzyliterową nazwę rozkazu. Warunki występujące w rozkazach mają skrót o maksymalnej długości czterech znaków, rozpoczynający się literą. Wykorzystanie tego pola nie jest obowiązkowe. W polu argumentu mogą wystąpić: liczba, adres bloku wykonawczego NAF, adres bloku pamięci Mqr (gdzie: N jest numerem stanowiska bloku w kasecie, A — adresem wewnętrznym w bloku, F — funkcją, M oznacza rozkaz do pamięci, q jest adresem w bloku pamięci, r — numerem bloku pamięci). Kolejne liczby w polu argumentu są oddzielone odstępem. Ze względu na ograniczenie zestawu do jednej kasety zrezygnowano z numeracji kaset w rozkazach adresowanych do bloków wykonawczych. Poza rozkazami akceptowanymi przez procesor wprowadzono instrukcję CONST umożliwiającą wstawienie dowolnej wartości do określonych miejsc programu wynikowego.

DZIAŁANIE TRANSLATORA

Translacja jest dokonywana w jednym przebiegu. Sterowanie procesem tłumaczenia odbywa się za pomocą instrukcji wierszy sterujących, które umożliwiają:

— drukowanie wersji źródłowej i wynikowej (w postaci dwójkowej i ósemkowej)
— wprowadzanie na perforator programu wynikowego w postaci ósemkowej, akceptowanej przez program ładujący
— wyprowadzanie programów w postaci plików bibliotecznych (identyfikowanych przez nazwę)
— dołączanie do programu uprzednio wygenerowanych plików.

Wczytany wiersz programu źródłowego jest tłumaczony na postać dwójkową, a następnie na postać ósemkową rozkazu. Po dokonaniu konwersji, jeśli w wierszach sterujących wystąpiła instrukcja wydruku, na drukarkę wyprowa-

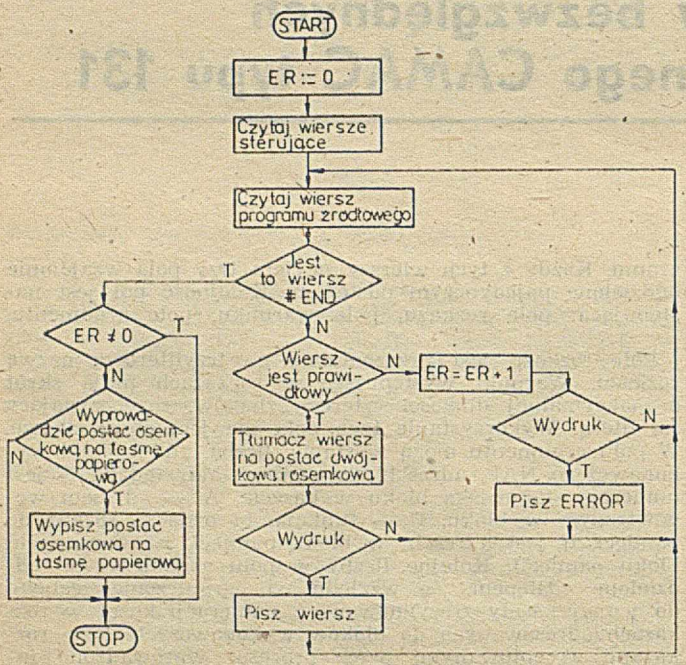
Mgr ZBIGNIEW REMPAŁA ukończył studia na Wydziale Elektrotechniki, Automatyki i Elektroniki Akademii Górniczo-Hutniczej w Krakowie. Pracuje na stanowisku technologa elektronicznego przetwarzania danych w Zakładowym Ośrodku ETO Zakładów Azotowych im. F. Dzierżyńskiego w Tarnowie.

Mgr inż. ROMAN RYGAŁ ukończył w 1981 r. studia na Wydziale Elektrotechniki Automatyki i Elektroniki Akademii Górniczo-Hutniczej im. St. Staszica w Krakowie, o kierunku — Elektronika. Od 1981 r. pracuje jako asystent w Instytucie Elektroenergetyki Politechniki Częstochowskiej, zajmując się wyładowaniami niezupełnymi w izolacji stałej.

dzany jest adres rozkazu w pamięci, postać źródłowa, dwójkowa i ósemkowa rozkazu. W ten sposób, wiersz po wierszu, tłumaczony jest cały program źródłowy.

Podczas translacji wykrywane i sygnalizowane są następujące błędy:

— nierozpoznany rozkaz
— nierozpoznany warunek
— nierozpoznany typ argumentu
— niedozwolona kombinacja elementów rozkazu.



Algorytm translatora

Po poprawnej translacji następuje wyprowadzenie postaci wynikowej programu na taśmę papierową zgodnie z poleceniem.

Informacja wejściowa do translatora jest wprowadzana z czytnika kart perforowanych. Przyjęto przy tym zasadę, że jeden rekord wejściowy stanowi jeden wiersz sterujący translacją lub jeden wiersz programu źródłowego.

* * *

Opisany translator można eksploatować na dowolnym zestawie komputerowym serii ODRA 1300 z pamięcią operacyjną o minimalnej pojemności 16 K słów. Podstawowy zestaw sprzętowy zawiera jednostkę centralną z konsolą operatorską, dwie jednostki pamięci taśmowej, czytnik i dziurkarkę taśmy papierowej.

Możliwości translatora są całkowicie wystarczające do wykonania oprogramowania procesora typu 131, pracującego w zestawie jednokasetowym. Translator usprawnia znacznie sam proces tworzenia oprogramowania, umożliwiając programiście skupienie się na koncepcji pisanych programów i eliminując konieczność przeliczania ich na postać ósemkową.

LITERATURA

[1] Gacek R., Gościński A., Rempała Z., Rygał R.: Analizator ATA-77 na bazie systemu CAMAC do analizy dwuparametrycznej (w przygotowaniu)

[2] Gecow A.: Język i asembler CAMASS-B2 na procesor autonomiczny CAMAC typ 131. Raport INR-1752/III/PM/B, Instytut Badań Jądrowych, Świerk, 1978

[3] PN-72/T-06530 — CAMAC. Blokowy system oprzyrządowania elektronicznego do pomiarów automatycznych i sterowania. Konstrukcja i organizacja logiczna

[4] Procesor autonomiczny typ 131. Instrukcja obsługi 155-IO. ZZUJ POLON, Zakład Aparatury Elektronicznej, Warszawa, 1980.

ALGORYTMY

Dwa algorytmy probabilistyczne

W poniższym artykule przedstawiamy dwa dalsze algorytmy probabilistyczne związane z grafami. Tym razem rozważać będziemy grafy niezorientowane bez cykli jednoelementowych, tzn. założymy, że zbiór wierzchołków $V = \{1,...,n\}$ oraz że zbiór krawędzi $E \subseteq \{\{v, w\}: v \neq w \wedge v, w \in V\}$.

Pokryciem krawędziowym grafu o $n = 2k$ wierzchołkach nazywamy zbiór k krawędzi $e_1,...,e_k$ ($e_i \in E$, dla $1 \leqslant i \leqslant k$), o własności: $e_1 \cup e_2 \cup \cdots \cup e_k = V$. Drogę i cykl Hamiltona oraz gęstość grafu definiujemy natomiast podobnie jak w przypadku grafów zorientowanych, por. [1, 4, 7]. Oba prezentowane algorytmy korzystać będą z funkcji select, opisanej dla grafów zorientownych w poprzednim artykule rubryki ALGORYTMY [7]. Aby dostosować ją do grafów niezorientowanych, wystarczy po prostu zamienić występujący w niej zapis krawędzi (v, w) na {v, w}, toteż nie będziemy tu przytaczać jej treści.

FUNKCJA PM [1]

Funkcja PM znajduje pokrycie krawędziowe grafu G. Wymagana gęstość G wynosi c*n*log(n). Wynikiem PM jest **true**, jeśli algorytm zakończy się sukcesem i wówczas zbiór reprezentowany przez zmienną globalną P zawiera pokrycie krawędziowe grafu G. Jeśli algorytm zakończy się porażką, to PM = **false**.

Liczba kroków wykonywanych podczas działania PM jest proporcjonalna do n*log(n), podczas gdy koszt najlepszego znanego algorytmu nieprobabilistycznego rozwiązującego podobny problem wynosi $0(n^{2 \cdot 5})$ [2]. Niech teraz G_N oznacza graf o gęstości N. Poniższe twierdzenie [1] charakteryzuje probabilistyczne własności algorytmu PM:

$\forall a > 0 \exists M, c \forall N > c \cdot \log:$

prawdopodobieństwo, że **PM**(G_N) = **true**, zanim funkcja select zostanie wywołana M*n*log(n) razy — wynosi $1-0(n^{-a})$.

A oto treść funkcji PM:

```
boolean procedure PM(G); graph G;
comment znajduje pokrycie krawędziowe grafu G;
begin integer ndp, v, u;
1 : P: ≠ ϕ
2 : if P zawiera wszystkie krawędzie z G
```

```
  then PM: = true; return
  else ndp: = najmniejszy wierzchołek nie występujący w
    P; go to 3
  fi;
3 : v: = select(ndp);
  if v = 0 then PM: = false; return; fi
  if v nie występuje w P then
    dołącz {ndp, v} do P; go to 2
  else
    u: = wierzchołek v taki, że {u, v}∈P;
    usuń {u, v} z P;
    dołącz {ndp, v} do P;
    ndp: = u;
    go to 3
  fi;
end PM
```

ALGORYTM UHC

Opiszemy najpierw funkcję UHP. Jej parametrami są:

— graf niezorientowany G o gęstości nie mniejszej niż c*n*log(n)

— wierzchołki s, t∈V.

Wynikiem UHP jest **true**, jeśli algorytm zakończył się sukcesem i wówczas zbiór reprezentowany przez zmienną globalną P zawiera drogę Hamiltona prowadzącą od s do t. Jeśli algorytm nie znajduje drogi to wynikiem jest **false**.

Algorytm UHP przedstawiony został w pracy [1]. Poprzednie algorytmy probabilistyczne, znajdujące cykle Hamiltona w grafach niezorientowanych, miały gorsze parametry niż UHP. Np. Perepelica w pracy [5] zaprezentował algorytm dla grafów o gęstości $c*n^{3/2}*\log^{1/2}(n)$, działający w czasie proporcjonalnym do n^2. Posa [6] wykazał, że prawie wszystkie grafy o gęstości c*n*log(n) zawierają cykle Hamiltona. Dowód Posy nie zawiera konkretnego algorytmu, ale na podstawie jego wyników można zbudować różne procedury rozwiązujące nasz problem. Niezależnie Koršunow podał algorytm dla grafów o gęstości c*n*log(n) działający w czasie $0(n^2\log^2(n)$. Natomiast koszt czasowy algorytmu UHP jest proporcjonalny do $n*\log^2(n)$.

Zauważmy, że problem znajdowania cykli Hamiltona w grafach niezorientowanych jest NP-zupełny, zatem funkcja UHP może mieć liczne praktyczne zastosowania. Oto jej treść:

```
boolean procedure UHP(G, s, t); graph G; integer s, t;
comment znajduje drogę Hamiltona od s do t w grafie G;
begin integer ndp, u, v;
1 : ndp: = s; P: = ∅
2 : comment P zawiera ścieżkę od s do ndp;
  if P zawiera każdy wierzchołek z G (z wyjątkiem t, je-
    śli s ≠ t) and {ndp, t} było usunięte z G
  then dołącz {ndp, t} do P;
    UHP: = true; return
  fi;
  v: = select(ndp);
  if v = 0 then UHP: = false; return; fi;
  if v = t then go to 2 fi;
  if v nie występuje w P
  then dołącz {ndp, v} do P;
    ndp: = v; go to 2
  fi;
  u: = najbliższy ndp sąsiad występujący w P;
  usuń {u, v} z P;
  dołącz {ndp, v} do P;
  ndp: = u;
  go to 2;
  end UHP;
```

Funkcja UHC znajduje cykl Hamiltona w grafie G o początku (i końcu) w wierzchołku s. Wynikiem jest **true**, jeśli cykl zostanie znaleziony i **false** w przeciwnym przypadku. Znaczenie zmiennej globalnej P jest takie, jak poprzednio.

```
boolean procedure UHC(G, s); graph G; integer s;
begin
  UHC: = UHP(G, s, s);
end UHC
```

LITERATURA

[1] Angluin D., Valiant L. G.: Fast Probabilistic Algorithms for Hamiltionian Circuits and Matchings. 9th ACM Symp. on Theory of Comp., Boulder, Colorado, USA, 1977

[2] Even S., Kariv O.: An $O(n^{2} \cdot ^{5})$ Algorithm for Maximum Matching in General Graphs. Proceedings of 16th IEEE Symp. on Found. of Comp. Science, 1975

[3] Koršunov A. D.: Solution of a Problem of Erdös and Rényi on Hamiltonian Cycles in Nonoriented Graphs, Soviet Math. Doklady 17, 1976

[4] Lipski W.: Kombinatoryka dla programistów. WNT, Warszawa, 1982

[5] Prepelica V. A.: On Two Problems from the Theory of Graphs, Soviet Math. Doklady 11, 1970

[6] Posa L.: Hamiltonian Circuits in Random Graphs, Discrete Math. 14, 1976

[7] Szałas A., Świrski Z.: Algorytm Angluina-Valianta. Informatyka, nr 4/1983.

ANDRZEJ SZAŁAS
ZBIGNIEW ŚWIRSKI

Zastosowania informatyki w przedsiębiorstwach w warunkach kryzysu i reformy gospodarczej

Komisja Informatyki Oddziału Wojewódzkiego PTE w Bydgoszczy, Instytut Badań Systemowych PAN w Warszawie oraz Zakład Informatyki Stosowanej ATR w Bydgoszczy organizują w dniach 28—30 września br. w Ciechocinku ogólnopolską konferencję „Stan oraz kierunki zastosowań informatyki w przedsiębiorstwach w warunkach kryzysu i reformy gospodarczej".

Celem konferencji jest dokonanie wszechstronnej analizy zastosowań informatyki w przedsiębiorstwach oraz ocena wpływu zjawisk kryzysowych na postęp w tej dziedzinie. Referaty będą prezentowały stan i pożądane kierunki przemian przy uwzględnieniu zasad i mechanizmów nowego systemu ekonomicznego. Rozeznanie zachodzących w przedsiębiorstwach przemian, przeprowadzone w oparciu o wyniki badań ankietowych Zakładu Informatyki Stosowanej ATR w Bydgoszczy oraz Komisji Informatyki OW PTE w Bydgoszczy, powinno pozwolić na określenie dalszych, bardziej efektywnych działań w warunkach modernizacji systemu zarządzania gospodarką narodową. Obrady koncentrować się będą na najbardziej adekwatnych do potrzeb krajowych, a jednocześnie postępowych kierunkach zastosowań informatyki w przedsiębiorstwach.

Koszt uczestnictwa w konferencji wynosi 5400 zł od osoby i obejmuje: zakwaterowanie i wyżywienie oraz materiały konferencyjne i pokonferencyjne. Materiały konferencyjne zostaną rozesłane uczestnikom w terminie do 15 września br. Organizatorzy przypominają, że podstawę prawną ponoszenia opłat konferencyjnych stanowi rozporządzenie Min. Finansów z dnia 12.7.65 (Monitor Polski nr 38, poz. 215). **Zgłoszenia uczestnictwa oraz opłat należy przesyłać do dnia 31.07.1983 r. pod adresem: Oddział Wojewódzki PTE, ul. Długa 34, 85-034 Bydgoszcz.**

Zastosowanie sterownika mikroprocesorowego MIKRO–80 w gospodarce

Jednodniowe seminarium na wymieniony temat, które odbyło się 27 października ub.r. w Poznaniu, zostało zorganizowane przez Ośrodek Badawczo-Rozwojowy Systemów Automatyki i Sekcję Metrologii, Automatyki i Mechaniki Precyzyjnej SIMP. W kilku referatach przedstawiono ogólne informacje dotyczące rozwoju systemu mikroprocesorowego MIKRO-80 oraz jego zastosowań w różnych gałęziach gospodarki.

Przegląd referatów

Modularny system mikroprocesorowy MIKRO-80 narodził się ze specjalizowanego sterownika przeznaczonego do pracy w tzw. magazynie wysokiego składowania. Szybko dostrzeżono uniwersalne możliwości tego urządzenia i dostosowano do ogólniejszych wymagań. Konstrukcję elektroniczną sterownika oparto na mikroprocesorze INTEL 8080A, a mechaniczną — na tzw. podwójnej eurokarcie i pośrednim złączu 84-stykowym. Magistrala systemu jest niestandardowa, a jej budowa przypomina MUBUS i MULTIBUS. Pojedyncza kaseta ma szerokość 19 cali i mieści 11 modułów oraz zasilacz (A. Wachowski).

Modele podstawowych modułów sterownika wykonano w 1979 roku. Obecnie produkuje się kilkanaście typów modułów funkcjonalnych, umożliwiających tworzenie różnorodnych zestawów (P. Wronek). Zasadnicze dane tych modułów przedstawiono w poprzednim numerze INFORMATYKI (str. 8). Obecne prace projektowe dotyczą m.in. modułów transmisji danych, współpracy z urządzeniami analogowymi, urządzeniami elektrycznymi większej mocy i komunikacji z operatorem (moduł monitora graficznego omówiono w INFORMATYCE nr 2/83, str. 24).

Prace rozwojowe w zakresie sprzętu prowadzone są w trzech podstawowych kierunkach, wyznaczonych zarówno przez zastosowania, jak i przez obecne tendencje w dziedzinie komputerowych systemów sterowania. Nowoczesna organizacja systemów sterowania obiektami przemysłowymi wymaga podziału właściwości funkcjonalnych między sterowniki procesu (z możliwością sterowania autonomicznego), stanowiska operatorskie i sterowniki komunikacyjne (Z. Pluciński).

Jedno z najciekawszych zastosowań sterownika — do automatyzacji różnych obiektów w hutnictwie (R. Janowski) — przedstawiliśmy w poprzednim numerze INFORMATYKI. Inne referowane prace dotyczyły sterowania ruchem ulicznym (M. Kaczmarek i in.), sterowania ruchem palet w magazynach (W. Grzybowski, B. Glajcher), automatycznego nadzorowania siłowni na statku (R. Szpakowski), kontroli i nadzoru obiektu klimatyzowanego (M. Kaczmarek, M. Jędrusiak). Omówiono je dokładnie w artykułach opublikowanych w Biuletynie Technicznym MERA, nr 10/1982 [1]).

We wszystkich wymienionych zastosowaniach, których zakres jest zadziwiająco szeroki, sterowniki MIKRO-80 charakteryzują się dużą funkcjonalnością. Występują w różnych konfiguracjach, złożonych zarówno z modułów typowych, jak i specjalizowanych. Wydaje się, że przygotowanie producenta jest solidne, a całe przedsięwzięcie ma ręce i nogi. Nie jest to, zresztą, jedyny system produkowany wspólnie przez OBR Systemów Automatyki i Zakłady Systemów Automatyki, przy współpracy z Instytutem Automatyki Politechniki Poznańskiej.

Na podstawie przedstawionych prac można z czystym sumieniem stwierdzić, że system MIKRO-80 ma wszelkie dane ku temu, aby zdobyć (może nawet powszechne) uznanie w kraju. Przemawiają za tym: dobra organizacja przedsięwzięcia, szeroki zakres zastosowań i możliwości techniczne systemu.

Uwagi krytyczne

Przy braku uzgodnionego, krajowego standardu, standardem stanie się system masowo produkowany. Jednak przedtem musi być on masowo kupowany, a do tego powinien spełniać podstawowe warunki techniczne i ekonomiczne, jak prostota, elastyczność i mały koszt.

Prostota systemu mikrokomputerowego przejawia się przede wszystkim w łatwości łączenia (zestawiania) modułów, łatwości programowania i konserwacji. Wydaje się, że system MIKRO-80 spełnia, dzięki modularności, jedynie warunek łatwości zestawiania. O łatwości konserwacji, której również sprzyja modularność, można powiedzieć dopiero wtedy, gdy sprawdzi się niezawodność i pozna warunki serwisu. Słabą stroną systemu wydaje się być jego oprogramowanie. Choć wszystkie zestawy są lepiej lub gorzej oprogramowane, niemniej nie widać śladu jakiejś jednolitej koncepcji.

Elastyczność systemu mikrokomputerowego zawiera się przede wszystkim w jego modyfikowalności i rozszerzalności. System MIKRO-80 ma ku temu doskonałe warunki. Warto jednak zwrócić uwagę, że spośród kilku równorzędnych krajowych systemów mikrokomputerowych wygra prawdopodobnie ten, który najwcześniej zapewni budowę standardowych zestawów rozłożonych przestrzennie (tworzenie struktur sieciowych). Trudno dostrzec, aby w systemie MIKRO-80 prace w tym kierunku rozwijano szczególnie intensywnie. Istotnym ograniczeniem elastyczności systemu w niektórych zastosowaniach mogą być wymiary mechaniczne i szybkość wykonywania operacji, określona przez właściwości magistrali.

Gdy mowa o kosztach zestawu, należy wziąć pod uwagę nie tylko ceny poszczególnych modułów, zasilaczy i kaset, które są stałe. Dużą rolę odgrywa koszt zaprojektowania, stworzenia dokumentacji i konserwacji zestawu. Dzięki walorom sprzętu (takim, jak modularność i niezawodność), koszty te można znacznie zredukować. Wtedy istotny staje się koszt początkowy zakupu zestawu: procesorów, podstawowych bloków funkcjonalnych, kaset i zasilaczy. Niestety, na temat kosztów sterownika MIKRO-80 nie ma żadnych danych.

Reasumując, utrzymanie energicznej polityki w zakresie projektowania, produkcji i zastosowań systemu MIKRO-80 może doprowadzić do zdominowania krajowego rynku przez ten system. Masowość produkcji, nawet przy słabościach technicznych czy niedoskonałości konstrukcji, stanowi najistotniejszy element w handlu urządzeniami informatycznymi. Kto wie, może mamy już do czynienia z przyszłym polskim standardem. Należy jednak ostrzec twórców systemu, że podstawowym jego mankamentem jest brak zgodności magistrali z jakąkolwiek normą międzynarodową.

JANUSZ ZALEWSKI

[1]) Pełny zbiór dziewięciu referatów opublikowali organizatorzy Seminarium (OBR Systemów Automatyki, ul. Czerwonej Armii 66/72, 61-807 Poznań, tel. 699-151)

Klub Użytkowników Mikroprocesorów

Od kilku lat polscy hobbyści-majsterkowicze konstruują na własną rękę domowe komputery, oparte na mikroprocesorach. Brakuje im kompletu materiałów, sprawdzonych wzorów, koniecznych funduszy. Sami najczęściej przecierają ścieżki w konstrukcji sprzętu i tworzą własne, bardzo skromne oprogramowanie.

Na razie liczbę zaawansowanych hobbystów można szacować na 100 do 1000. W nadchodzących latach liczba ta może znacznie wzrosnąć, nawet do kilkudziesięciu tysięcy, zwłaszcza jeżeli powstaną koła zainteresowań w szkołach średnich.

Zapowiedziano poza tym produkcję polskiego mikroprocesora opartego na układach o wysokim stopniu scalenia. Egzemplarze serii próbnej wypuścił już Instytut Technologii Elektronowej. Rysuje się też możliwość stosowania procesorów doskonalszych, 16-bitowych, dzięki ewentualnej pomocy Towarzystwa Konsultantów Polskich.

Ze wstępnej rozmowy z wiceprezesem Polskiego Towarzystwa Informatycznego, dr. Waligórskim, wynika, że PTI gotowe jest traktować ewentualny klub użytkowników mikroprocesorów jako swoją sekcję i służyć mu pomocą i kontaktami.

Proponuję zatem nie zwlekać dłużej z założeniem Klubu Użytkowników Mikroprocesorów. Poniżej przedstawiam koncepcję takiego klubu. Zainteresowane osoby lub instytucje proszone są o nadsyłanie — pod adresem niżej podpisanego autora lub redakcji INFORMATYKI — zgłoszeń lub propozycji, z podaniem uwag do przedstawionej koncepcji Klubu, a zwłaszcza swej roli w jego przyszłej działalności. Zgłoszenia te pozwolą zorientować się, na ilu organizatorów, członków i sympatyków będzie można liczyć.

KONCEPCJA KLUBU

Celem działania Klubu byłoby możliwie szybkie wprowadzenie mikroprocesorów do powszechnego użytku, w nadziei usprawnienia działań w wielu dziedzinach życia społecznego w Polsce. Cel ten można by osiągnąć za pomocą:

- wymiany informacji na temat:

— produktów służących do konstruowania urządzeń

— schematów i wzorów do naśladowania

— oprogramowania

— poszczególnych użytkowników systemów mikroprocesorowych

— światowych osiągnięć w tej dziedzinie

- spotkań i szkoleń
- umożliwienia najmłodszemu pokoleniu dostępu do sprzętu i wiedzy
- wypracowywania wzorów i standardów rozwiązań urządzeń i oprogramowania.

PROPONOWANE PRZEDSIĘWZIĘCIA

- Wydawanie corocznych spisów adresów i telefonów instytucji oraz osób zrzeszonych w klubie, z podaniem przedmiotu ich działalności oraz typu i rodzaju zastosowania mikroprocesorów.
- Wydawanie periodycznych ofert produktów i usług — na koszt ogłaszającego — rozsyłanych do wszystkich członków Klubu. Jako aneks — ogłoszenie POTRZEBNA POMOC — na temat rozwiązania konkretnego problemu.
- Wysyłanie abonentom indywidualnym szczegółowych prospektów przedsięwzięć Klubu.
- Wysyłanie — na podstawie zamówień i przedpłat — informatorów, podręczników, schematów, opracowań.
- Utworzenie BIBLIOTEKI OPROGRAMOWANIA, umożliwiającej wymianę programów.
- Założenie FUNDUSZU PROMOCYJNEGO. Na ten fundusz mogą się składać osoby prawne, będące członkami wspierającymi.
- Zorganizowana sprzedaż wysyłkowa kompletów „Zrób Sam" dla członków kół zainteresowań (np. w szkołach).
- Planowa działalność wydawniczo-popularyzatorska.
- Ewentualne inicjowanie produkcji niezbędnego sprzętu lub prac związanych z oprogramowaniem.

NAZWA

Należałoby przyjąć dla Klubów łatwą do zapamiętania nazwę, budzącą przyjemne skojarzenia. Alternatywą nazwy KUM mógłby być: KREM (Klub Realistycznych Entuzjastów Mikroprocesorów) lub MIKROKLUB.

ANDRZEJ DROŻNIAK

tel. dom. 41-26-01
tel. pr. 25-16-21 w. 195 (B.P. „CEWOK")

GIEŁDA INFORMACJI

TABULATOR

ZWCh „STILON" oferuje TABULATOR — zestaw podprogramów dla komputerów ODRA, który ułatwia pisanie programu w przypadku sporządzania wydruku w postaci małych tabelek (odcinków).

Dla maksymalnego wykorzystania powierzchni papieru i jednocześnie zachowania przy wydruku odpowiedniej sekwencji, programista zmuszony jest organizować w pamięci odpowiednie tablice i dla każdego wiersza w skomplikowany sposób kompletować informacje do wydruku. Trudności te całkowicie usuwa TABULATOR. Dzięki niemu programista przyjmuje za stronę jeden odcinek i każdy z nich wyprowadza w sposób sekwencyjny, nie martwiąc się o kompletowanie całych stron wydruku. Zestaw ten powinien zachęcać również do lepszego wykorzystania papieru w przypadku typowych tabulogramów, dających się zmieścić w większej liczbie na jednej stronie.

Kontakt merytoryczny: Romuald Dziubiński; ZWCh „STILON", ul. Walczaka 25, 66-407 Gorzów Wielkopolski, tel. 72-2335.

System F-K

Ośrodek obliczeniowy przy Częstochowskich Zakładach Przemysłu Lniarskiego WARTA w Częstochowie oferuje użytkownikom MERY 400 system finansowo-księgowy. System ten można eksploatować jako niezależny całościowy system ewidencji. Charakteryzuje się on prostotą wprowadzania danych źródłowych i obsługi oraz szczegółową kontrolą danych wejściowych, a także nie wymaga zmian w dotychczasowej dokumentacji.

System zapewnia informacje potrzebne do analizy kont oraz do sprawozdawczości, pozwala na: uzyskanie tabulogramów użytkowych i kontrolnych, zakładanie i aktualizację indeksu kont wraz z nazwami, listuje dokumenty błędne. System ten może być stosowany przez każde przedsiębiorstwo posiadające MERĘ 400 w konfiguracji standardowej.

Informacji udziela: kierownik Zakładowego Ośrodka Obliczeniowego, inż. Marian Kordys; Częstochowskie Zakłady Przemysłu Lniarskiego WARTA, 42-200 Częstochowa, ul. Krakowska 45, tel. 426-31 do 33 wew. 289, 332, teleks 037226 pl.

SPRZĘT

WZRS „Samopomoc Chłopska" w Szczecinie posiada na zbyciu sprawne technicznie maszyny:

— sprawdzarkę „Soemtron" (nr fabr. 53137, nr inwentarza HPL ST 110)

— dziurkarkę „Soemtron" 415" (nr fabr. 15814, nr inwentarza HPL ST 111).

Adres: WZSR „Samopomoc Chłopska" w Szczecinie, Zakład Obrotu Artykułami Przemysłu Lekkiego, Dział Techniczny i Zaopatrzenia Materiałowego, ul. Pomorska 53, 70-958 Szczecin, tel. 613-149.

Gdy komputer będzie słyszał

Idea rozpoznawania mowy przez komputer fascynowała ludzi od dawna. Pierwsze prace naukowe poświęcone temu zagadnieniu prowadzono już w roku 1943! Wtedy to właśnie M. I. Mjasnikow opublikował rozprawę o analizie „obiektywnego rozpoznawania dźwięków głosu", a rok później J. J. Dreyfus-Graf rozpoczął swoje badania dotyczące sonografu (zakończone w latach 1966—1971 konstrukcją aparatów nazwanych fonetografami).

W 1947, sporządzono pierwszy graficzny obraz dźwięku (rozkład częstotliwości w funkcji czasu).

I tak osiągnięto w 1952 pierwszy sukces — rozpoznanie dziesięciu cyfr wypowiadanych przez jednego człowieka; w 1960 powtórzono tę próbę stosując metody numeryczne.

W 1968 umiano już rozpoznać pojedyncze słowa (od 80 do 500 wyrazów), a w 1969 otrzymano pierwsze pozytywne rezultaty w rozpoznawaniu mowy ciągłej.

Liczni badacze żywili nadzieję, że w każdym słowie można wydzielić kilka stanów fizycznych dobrze odróżnialnych. Sprawa okazała się jednak bardziej skomplikowana, stąd tyle zniechęcenia i częste rezygnacje z nowych prób badawczych.

Korzystając z pozytywnych wyników badań nad rozpoznawaniem pojedynczych słów, firma THRESHOLD wyprodukowała w 1972 pierwszy aparat przeznaczony na sprzedaż — VIP 100, który potrafił odróżnić 32 pojedyncze słowa. Od tej pory zaczęły pojawiać się na rynku najróżniejsze aparaty tego typu, wykorzystujące szeroką gamę algorytmów o mniejszej lub większej elastyczności, ale zawsze słowa musiały być w nich odseparowane od siebie.

W 1971 Amerykański Departament Obrony przeznaczył 15 mln dol. na badania i rozwój systemów zdolnych do rozpoznania całych zdań wypowiadanych w sposób ciągły przez dowolne osoby. Specjalny program SUR (Speech Understanding Research) zakładał prace nad zbiorem 1000 słów i dopuszczał 10% stopę błędów. Doprowadził on do powstania licznych systemów: Harpy i Hearsay II (Carnegie Mellon University), HWIN (Bolt Beranek Newman), SDC (System Development Corp.). Okazało się, że największą trudność sprawia dowolny dobór mówiącego. Czas odpowiedzi najszybszego systemu (testy przeprowadzono z pięcioma osobami) sięgał kilku minut, a nawet kilku godzin!

Mimo wielu niepowodzeń największe firmy światowe kontynuują obecnie prace nad rozpoznawaniem mowy: IBM, BELL, TEXAS INSTRUMENTS, CARNEGIE-MELLON UNIVERSITY — w Stanach Zjednoczonych; NTT, NEC, HITACHI — w Japonii oraz CNET, CRIN, CEA, ENSERG, LIMSI — we Francji.

PO CO ROZPOZNAWAĆ MOWĘ

Cel rozpoznawania mowy jest jasny: przyspieszyć wprowadzanie informacji do komputera. Mowa ludzka jest najbardziej spontaniczną i bogatą formą kontaktu pomiędzy człowiekiem i maszyną, ale niestety — również najtrudniejszą.

Potencjalne możliwości zastosowań systemów rozpoznawania mowy w czasie rzeczywistym możemy podzielić na pięć dużych grup:

- sterowanie procesami w przemyśle (a także w wojsku i lotnictwie)
- zbieranie danych i zdalne przetwarzanie transakcji (np. handel i banki)
- zarządzanie i dostęp do baz danych
- dostęp do komputera
- nauczanie za pomocą komputera.

*

W lotnictwie Amerykanie nazywają ludzki głos „trzecią ręką". W niektórych samolotach naddźwiękowych przeznaczonych do szkolenia, piloci sterują wszelkimi przyrządami wyłącznie swoim głosem. Automatyczne rozpoznawanie mowy pozwoliło uwolnić ręce operatora, umożliwiając mu w ten sposób dodatkowe ruchy.

Na lotniskach amerykańskich wykorzystuje się powszechnie ludzki głos do sterowania rozdziałem bagaży podróżnych: urzędnik wymienia nazwę docelowego miasta, a maszyna kieruje walizki w odpowiednich kierunkach.

Również administracja usług pocztowych w USA zdecydowała się wykorzystać do swoich celów system sortowania paczek — Continous Speech Recognition System firmy DIALOG.

We Francji, w Laboratoire d'Automatique firmy BESANCON, prowadzone są próby sterowania za pomocą głosu robotami stosowanymi w medycynie.

Wszędzie tam, gdzie praca wymaga ciągłej uwagi operatora (np. w przypadku procesów chemicznych) rozważa się możliwość sterowania głosem. Kierowanie za pomocą ustnego polecenia stosuje się także w licznych grach i zabawkach. Opracowuje się też „głosowe" zamki, radia i telewizory, domowe komputery.

*

Firma LIMSI opracowała system pomocy w nauczaniu języków programowania. Student dyktuje komputerowi program, przy czym jedno słowo odpowiada fragmentowi instrukcji. Komputer wychwytuje wszystkie błędy syntaktyczne i sygnalizuje je za pomocą syntezatora mowy Icophone, który w przypadku błędu określa jego rodzaj lub wskazuje zbiór słów poprawnych.

*

Japońska firma MATSUSHITA zapowiedziała wprowadzenie w bieżącym roku na rynek fonicznej maszyny do pisania. Również firmy IBM, XEROX i UNIVAC prowadzą liczne prace w tej dziedzinie.

W Grenoble (Francja), firma ENSERG realizuje prace nad głosowym dostępem do baz danych i obecnie funkcjonuje już mały system czasu rzeczywistego, zbudowany na mikroprocesorze 8085. Wykorzystuje on kilkadziesiąt słów i wykonuje polecenia typu: „podaj wolne miejsca lotu do Paryża" czy „podaj ceny biletów lotu".

W kryminalistyce automatyczne rozpoznawanie „wokalnych podpisów" posłuży do walki z bandytyzmem, np. eksperymentalny system „Auros" (Automatic Recognition of Speakers) firmy PHILIPS FORSCHUNGS (w Hamburgu) realizuje bazę danych zawierającą 5000 krótkich zdań typu „Nazywam się Nemo".

Centrum CNET stworzyło system „Phonex". Każdy abonent tego systemu ma własny „bank foniczny" o pojemności 1 M bajta, zabezpieczany przez odpowiednie hasło dostępu i identyfikowany przez numer. Abonent może wprowadzać do systemu dźwiękową wiadomość przeznaczoną dla dowolnych abonentów lub wysłuchać wiadomości skierowanej do niego przez innych.

*

Niemożność lub ogromne trudności kontaktu ze światem zewnętrznym powodują znaczne opóźnienia w rozwoju ludzi niepełnosprawnych. Pojawienie się nowych technik syntezy i rozpoznawania mowy stwarza im możliwości pomocy w wymianie myśli czy samodzielnego dostępu do informacji.

Edukacja dzieci źle słyszących, sterowana ludzkim głosem, jest od 1974 tematem badań uniwersytetu w Nancy. Opierając się na podstawowej tezie, że dziecko głuche najczęściej tylko dlatego nie mówi, że nie słyszy — opracowano wizualny system „Sirène" zrealizowany na mikrokomputerze MITRA 125.

Idea pomocy wizualnej nie jest nowa i pochodzi od Bella, który za pomocą „widocznego słowa" nauczał głuche dzieci. Stosowanie narzędzi informatycznych ułatwia pracę foniatrom, umożliwiając przekazywanie dźwię-

ków wypowiadanych przez uczniów, a następnie ich kompleksową analizę i ustalenie metod nauczania, zapewniających na każdym etapie maksymalne wyniki.

Opracowano trzy zestawy zabaw pomagających dzieciom w kontroli stabilności napięcia strun głosowych (dziecko musi odtworzyć melodyczny zarys informacji), parametrów częstotliwościowych, które decydują o barwie głosu, indywidualnej charakterystyce wymawianych dźwięków, a w końcu zasobu słów i elementów zdań.

Na uniwersytecie w stanie Carolina (USA) utworzono z myślą o upośledzonych studentach specjalne laboratorium chemiczne, obsługiwane przez system zbudowany na bazie dwóch mikroprocesorów Z 80, wyposażony w 147 K słów pamięci. Mikroprocesor nadrzędny (ang. master) zbiera dane o pomiarach laboratoryjnych oraz przeprowadza obliczenia konieczne do ich przetwarzania, natomiast rola procesora podrzędnego (ang. slave) sprowadza się do konwersacji ze studentem. Jedną z wersji systemu „Ultra" (Universal Laboratory Training and Research Aid) przeznaczono dla studentów niewidomych, w którym mikroprocesor podrzędny pracuje jako terminal wyłącznie z wyjściem głosowym. Inna wersja tego systemu, sterowana z kolei głosem (układ Volterm firmy INTERSTATE ELECTRONICS rozpoznaje 100 słów) wyposażona jest w ekran przeznaczony do wyświetlania informacji wyjściowych.

Dzięki układom sprzężenia z obiektem, system „Ultra" może współpracować z licznymi przyrządami laboratoryjnymi: pH-metrami, spektrofotometrami, chromatografami, wagami analitycznymi, czujnikami temperatury itp.

TECHNIKI ROZPOZNAWANIA MOWY

Podstawową jednostką przetwarzania w podejściu całościowym jest niepodzielne słowo, którego sygnał porównuje się bezpośrednio z komputerowymi opisami. Metoda taka pozwala na uniknięcie efektów wzajemnego wpływu dźwięków wewnątrz słowa i chociaż w chwili obecnej ogranicza zarówno liczbę dopuszczalnych słów, jak i liczbę mówiących, to właśnie na niej opiera się budowa większości sprzedawanych dzisiaj systemów.

Podejście analityczne zakłada wykrywanie i identyfikację elementarnych składników słowa: fonemów, fonatomów i sylab, podstawowych rozpoznawalnych jednostek. Dzięki tej metodzie, wystarczy utworzyć w pamięci komputera opis charakteryzujący podstawowe cechy jednostek bazowych.

Wyróżnia się trzy kategorie systemów:

- oparte na rozpoznawaniu fonemów (SDC, MYRTILLE, KEAL)
- wykorzystujące metodę całościową (DRAGON, HARPY, NEC, ENSERG)
- stosujące równocześnie metodę całościową i analityczną (BBN, HEARSAY II)

Systemy wykorzystujące wyłącznie metodę całościową lub wyłącznie metodę analityczną mogą być wstępujące lub zstępujące. System nazywamy wstępującym, jeśli analiza lingwistyczna jest podrzędna w stosunku do rozpoznawania słów. W przeciwnym przypadku system jest zstępujący, a ograniczenia lingwistyczne sterują rozpoznawaniem.

Z punktu widzenia sposobu korzystania z wiadomości, wyróżnia się ponadto systemy aktywne i pasywne. W systemie pasywnym (KEAL) rozpoznawanie odbywa się szeregowo: analiza akustyczno-fonetyczna, analiza leksykalna, analiza syntaktyczna itd., natomiast system aktywny (HARPY, HEARSAY II, BBN) charakteryzuje równoległe wykonywanie poszczególnych analiz.

Po okresie bardzo intensywnej pracy nad projektem SUR, amerykańscy naukowcy pracują dzisiaj znacznie spokojniej i ukierunkowują swoje systemy przede wszystkim na potrzeby przemysłu (HARPY) oraz handlu (VIP 100). Do prac nad rozpoznawaniem mowy włączyła się również przed kilkoma laty największa z komputerowych potęg — firma IBM.

Japonia jest jedynym obok USA krajem, w którym urządzenia do rozpoznawania mowy występują już normalnie w sprzedaży. DP 100 firmy NIPPON ELECTRIC COMPANY (NEC) uznaje się za jeden z najlepiej dopracowanych aparatów dostępnych na rynku. Zbudowany jest on na bazie trzech mikroprocesorów, z których jeden służy jako interfejs operatora, a pozostałe dwa uczestniczą w rozpoznawaniu słów.

* * *

Prace badawcze nad rozpoznawaniem mocy znajdują się obecnie na etapie ograniczonego rozumienia, tzn. przypuszcza się, że sam sygnał akustyczny nie zawiera wszystkich potrzebnych informacji niezbędnych do zrozumienia wiadomości i że należy sięgnąć do innych źródeł wiedzy (fonetycznych, prozodycznych, syntaktycznych, semantycznych, pragmatycznych) niezbędnych również do wychwycenia wszystkich słów, które zawiera sygnał.

Następny etap prac zakłada zrozumienie i rozpoznanie mowy naturalnej i potocznej, praktycznie bez żadnych ograniczeń nakładanych na syntaktykę i słownik języka. Trzeba będzie wówczas zidentyfikować przynajmniej 10 tys. słów.

Oprac. MARIANNA SOBCZYK
na podstawie „Minis et Micros"
z 20.09.1982

Seria HP 9000

Szybki postęp w dziedzinie technologii półprzewodnikowej, jakiego ciągle jesteśmy świadkami, spowodował, że coraz trudniej klasyfikować nowe wyroby do tradycyjnie ustalonych grup. Na początku tego roku firma HEWLETT-PACKARD rozpoczęła sprzedaż w Europie Zachodniej maszyn serii 9000, która właśnie łamie dotychczas przyjęte podziały. Z jednej strony parametry użytkowe wskazują, że mamy do czynienia z minikomputerem (i to dosyć wysokiej klasy), z drugiej zaś technologia wykonania pozwala zaliczyć ten model do grupy mikrokomputerów.

Rodzina HP 9000 jest przeznaczona przede wszystkim do zastosowań naukowo-technicznych i w wersji podstawowej przybiera postać komputerów stołowych (ang. desk top). Jakkolwiek obecnie HEWLETT-PACKARD nie ma zbyt wielu konkurentów w dziedzinie maszyn tego typu, to jednak przedstawiana dziś seria jest owocem ponad 6 lat prac badawczych i ok. 100 mln dolarów poświęconych na ten cel.

Dziewięć modeli

Na rodzinę HP 9000 składają się maszyny trzech poziomów, zależnie od liczby procesorów w jednostce centralnej: HP 9000 seria 500 (1 procesor) HP 9000 seria 600 (2 procesory) oraz HP 9000 seria 700 (3 procesory). Z kolei, na każdym poziomie występują trzy wersje: model 20 — zintegrowany w jedno stanowisko pracy kompu-

ter stołowy, model 30 — panel przeznaczony do zamontowania w stojaku i model 40 — miniszafka stanowiąca trzon najbardziej rozbudowanych konfiguracji. Obecnie na rynku dostępne są tylko 3 wersje serii 500 (jednoprocesorowej).

Dodatkowo poszczególne konfiguracje HP 9000 mogą różnić się doborem urządzeń zewnętrznych. Producent oferuje w tym przypadku: monitory ekranowe graficzne (jedno- i wielobarwne), klawiatury (typu Qwerty lub Azerty), drukarki (np. cichobieżna drukarka termiczna), jednostki pamięci na dyskach elastycznych (o pojemności 270 K bajtów) oraz jednostki pamięci dyskowej typu Winchester (najmniejsza pojemność 10 M bajtów). Istnieje również możliwość rozbudowy pamięci operacyjnej od 256 K do 2,5 M bajtów (w modułach po 256 K bajtów).

Technologia NMOS III

Najważniejszym osiągnięciem firmy HEWLETT-PACKARD, zastosowanym w omawianej rodzinie komputerów, jest skonstruowanie układów scalonych VLSI w oparciu o technologię NMOS III. Pozwoliła ona uzyskać trzy do ośmiu razy większą gęstość liniową rozmieszczenia elementów w układach, w porównaniu do dotychczas najlepszych rezultatów w tej dziedzinie (np. układy firmy INTEL wykonane technologią NMOS II). Dzięki temu udało się zmieścić w jednym układzie nawet 70 razy więcej elementów. Wyższy stopień scalenia układów, a więc redukcja ich liczby w całym komputerze, prowadzi z kolei do zwiększenia jego niezawodności.

Do budowy HP 9000 użyto tylko pięciu rodzajów podstawowych układów. Pierwszy z nich to 32-bitowy mikroprocesor centralny, odpowiadający 450 tys. tranzystorów (a 10 tys. już kwalifikuje układ do grupy układów wielkiej skali integracji LSI). Posiada on wbudowaną pamięć sterującą o pojemności 9 K słów 36-bitowych, a jego podstawowy cykl wynosi 55 nanosekund, co odpowiada prędkości 1 miliona operacji/s. Komunikację zapewniają 32-bitowe magistrale danych i adresów o prędkości 36 M bajtów. Ten ostatni parametr decyduje o tym, że mamy do czynienia z prawdziwym 32-bitowym procesorem (zapowiadany przez firmę INTEL mikroprocesor iAPX 432, jest 32-bitowy tylko z punktu widzenia oprogramowania, a nie z punktu widzenia architektury sprzętu). Przyjęte rozmiary magistrali pozwalają adresować bezpośrednio (przynajmniej teoretycznie) pamięć o pojemności do 500 M bajtów i nie ponosić zbyt wielkich strat czasu na przesłania podczas wykonywania obliczeń na liczbach 32- i 64-bitowych, jak dzieje się w „fałszywych" procesorach 32-bitowych.

Układ IOP jest wyspecjalizowanym procesorem wejścia-wyjścia stanowiącym odpowiednik 240 tys. tranzystorów. Pracuje w oparciu o własne mikroprogramy, przechowywane w pamięci o pojemności 4,5 K słów 38-bitowych. IOP może zarządzać ośmioma oddzielnymi kanałami i zapewnia transmisję z prędkością 6 M bajtów/s.

Trzeci układ — pamięć typu RAM — charakteryzuje się największą, niespotykaną dotąd, gęstością elementów — 600 tys. tranzystorów na jednej płytce. Dzięki temu osiągnięto, również po raz pierwszy, pojemność pamięciową 128 K bitów. Dotychczas najbardziej pojemne układy pamięci miały 64 K, zaś w powszechnym użytku występują kostki zaledwie 16 K bitowe. Mimo tak dużej pojemności pamięć ta jest bardzo szybka — cykl wynosi bowiem tylko 110 nanosekund.

Układ następny to sterowanie pamięcią operacyjną. Umożliwia on zarządzenie 256 K bajtami pamięci RAM, a także wykrywanie i poprawianie błędów oraz testowanie układów pamięci po spadku napięcia zasilającego. I wreszcie ostatni, piąty układ to zegar dostarczający dwu oddzielnych sygnałów o częstotliwości 18 MHz.

Wszystkie wymienione układy mają wbudowany specjalny system odprowadzania ciepła o nazwie „Finstrate". System ten zapewnia możliwość chłodzenia komputera za pomocą niewielkiego wentylatora.

Osiągnięty przez firmę HEWLETT-PACKARD stopień upakowania elementów pozwala zbudować 32-bitową maszynę — z pamięcią operacyjną o pojemności 1 M bajta — zaledwie z 98 układów scalonych. Zbudowanie jej z układów NMOS II wymagałoby użycia ok. 800 układów (dostępne obecnie na rynku komputery 32-bitowe zawierają przeciętnie ok. 5000 układów).

Oprogramowanie

Użytkownicy maszyn z rodziny HP 9000 mogą wybrać jeden z dwóch systemów operacyjnych: HP-UX (wersja słynnego systemu UNIX opracowanego w BELL LABORATORIES) lub BASIC. Pierwszy z nich przeznaczony jest dla zawodowych informatyków, drugi zaś — dla pozostałych użytkowników (jak wspomniano na wstępie, HP 9000 może być traktowany również jako komputer osobisty).

HP-UX dopuszcza jednoczesną pracę z wielu terminali, natomiast BASIC przewiduje wyłącznie pracę jednostanowiskową. Liczne adaptacje dokonane przez firmę HEWLETT-PACKARD w stosunku do oryginalnego UNIXA miały na celu zapewnienie kompatybilności z dotychczas stosowanymi systemami HP, np. systemem zarządzania bazą danych IMAGE, czy systemem GRAPHICS. HP-UX zawiera kompilatory języków PASCAL, FORTRAN i C, programy współpracy z siecią typu ETHERNET oraz monitor podziału zasobów SRM do realizacji współpracy wielu maszyn HP 9000.

Z kolei najważniejszą cechą systemu BASIC jest jego całkowita kompatybilność z poprzednimi wersjami 16-bitowymi (szczególnie z serią HP 9800 zwaną obecnie HP 200), co zapewnia wymianę sprzętu na nowy bez typowych kłopotów związanych z przerabianiem oprogramowania. Dla zwiększenia mocy obliczeniowej nowy system BASIC pozwala na pracę wieloprogramową (ale z jednej konsoli) oraz zawiera oprócz tradycyjnego interpretera języka BASIC, również jego kompilator. Oczywiście i w tym przypadku można korzystać z sytemu IMAGE i związanego z nim języka zapytań QUERY.

Ponieważ adresatami systemu HP 9000 są przede wszystkim wykonawcy prac projektowych, producent oferuje również specjalistyczne pakiety oprogramowania, np. HPspice (symulacja układów elektronicznych), HP FE II (analiza metodą elementów skończonych), czy HPdesign (konstrukcje mechaniczne).

* * *

Biorąc pod uwagę wysokie walory prezentowanego sprzętu, jego ceny nie są zbyt wygórowane. Na przykład, HP 900 seria 500 model 20 wyposażony w zintegrowane stanowisko pracy, pamięć operacyjną o pojemności 1 M bajt, pamięć na dysku elastycznym o pojemności 270 K bajtów, pamięć dyskową typu Winchester o pojemności 10 M bajtów, drukarkę termiczną, systemy operacyjne BASIC i HP-UX, kompilatory FORTRANU i PASCALA oraz bibliotekę graficzną i monitor ekranowy (jednobarwny) kosztuje we Francji ok. 450 tys. franków (ok. 64 tys. dolarów), zaś w tej samej konfiguracji, lecz z monitorem ekranowym wielobarwnym — ok. 580 tys. franków (ok. 83 tys. dolarów).

Oprac. MaS
na podstawie
„01 INFORMATIQUE"
nr 724 z 22 listopada 1982 r.

Niekonieczna „Praktyka programowania"

Ostatnio na naszym rynku ukazało się drugie już wydanie książki [1]), której oryginalny tytuł w tłumaczeniu brzmi: „Styl programowania, projektowanie, sprawność, uruchomianie i testowanie programu". Można powiedzieć, że jest to także najkrótsze streszczenie tej pracy. Książka zawiera bowiem pięć rozdziałów poświęconych wymienionym elementom praktycznego programowania.

Pierwszy rozdział, o stylu programowania, to — moim zdaniem — najsilniejszy punkt książki. Zebrano w nim rzetelną wiedzę o pewnych najprostszych, ale nierzadko pomijanych, czynnikach decydujących o czytelności programu. Mamy tu więc wykład o stosowaniu komentarzy, nazywaniu zmiennych, używaniu skrótów oraz redagowaniu tekstów.

Autor podsumowuje swoje kolejne uwagi maksymami programistycznymi. Wśród nich znalazła się wyjątkowo celna: *„Programy mają być czytane przez ludzi"*. Szkoda, że nie cała książka ją uwzględnia.

W rozdziale drugim omówiono podstawowe zasady procesu projektowania, akcentując metodę projektowania zstępującego i programowanie strukturalne. Autor zaleca także (a — co ważne — rzadko jest to praktykowane) sięganie do dostępnych zbiorów algorytmów, jak również do istniejących często bibliotek procedur. Ogólną koncepcję prostoty wyraża maksyma *BUZI* czyli *„Bez Udziwnień Zapisuj, Idioto"* (brawo tłumacz — w oryginale zasada ta brzmi: *KISS — Keep It Simple, Stupid).*

Trzeci rozdział jest poświęcony sprawności (inaczej — efektywności) programów. I choć zaczyna się od bardzo słusznej maksymy *„Gdy coś nie działa, jego sprawność nie ma żadnego znaczenia"*, to jednak w dalszej części przyjęto za właściwe także potencjalnie niebezpieczne metody usprawniania. Przykładem może być propozycja optymalizacji wykorzystania pamięci poprzez tzw. równoważenie zmiennych — sposób „wyklęty" przez znakomitą większość informatyków (jako zmniejszający niezawodność). Natomiast podany na str. 129 przykład oszczędzania czasu i pamięci za pomocą nadawania zmiennym wartości początkowych, podczas deklaracji w języku PL/I — nie zawsze jest prawdziwy (dla zmiennych z atrybutem AUTOMATIC kompilator i tak wygeneruje odpowiednie instrukcje podstawienia).

W rozdziale czwartym zebrano możliwie wszystko, co składa się na uruchamianie programu. Trudno doszukać się w nim treści odkrywczych, zwłaszcza, że autor dedykuje swoją książkę osobom potrafiącym programować. Wiele zawartych w tym rozdziale zaleceń, to konsekwencja poprzednich etapów prac nad programem i nie wydaje się konieczne powtórne ich przytaczanie.

Ze szczególną rezerwą należy potraktować rozdział piąty — o testowaniu. Autor nawet sam sobie przeczy, pisząc początkowo o niemożności przetestowania wszystkich sytuacji w programie, a następnie — nawołując do testowania wszystkich dróg. Swoją drogą, zadziwiająca jest żywotność mitu o testowaniu ścieżek.

I wreszcie — rozdział szósty, czyli 101 zadań do programowania, które, wraz z ćwiczeniami kończącymi pozostałe rozdziały pozwalają zastosować podane zasady w praktyce. Niestety, zbyt wiele z nich posiada odwołania do trudno dostępnej w Polsce literatury (szczególnie w mniejszych ośrodkach).

Jak pisze w przedmowie autor, *„książka jest przeznaczona dla osób, które potrafią już programować i chcą osiągnąć biegłość w programowaniu"*. Wydaje się jednak, że część przedstawionego materiału (głównie dwa pierwsze rozdziały) mogłaby stanowić uzupełnienie początkowego nauczania, programowania (wszak łatwiej nauczyć dobrego stylu niż wykorzenić zły). Natomiast pozostała część, która zawiera treści dość dyskusyjne, wymaga od czytelnika pewnego przygotowania, by mógł zachować krytyczne spojrzenie na przedstawione fakty. Dlatego też uważam, że głównymi jej odbiorcami powinni być ludzie uczący informatyki, a nie uczący się sami.

Można też zarzucić autorowi pewną nienowoczesność omawianego materiału (co czyni w pewnym miejscu tłumacz) — wiele problemów dotyczy przestarzałych koncepcji języków FORTRAN, COBOL, PL/I, zaś należy wymagać, by narzędzia występujące w takim języku jak PASCAL nie były traktowane w roku 1978 jako eksperymentalne nowinki.

Książka traktuje informatykę jako rzemiosło. Niewiele tego typu pozycji pojawia się w Polsce, niemniej radziłbym Czytelnikom raczej zapoznanie się z inną książką Biblioteki Inżynierii Oprogramowania WNT, z „Projektowaniem niezawodnego oprogramowania" G. J. Myersa, która porusza podobne problemy, a nie posiada tak istotnych wad, jak praca Van Tassela. Jego książkę uznałbym w sumie za „miłą, lecz niekonieczną". Dziwi mnie przy tym fakt, że wydano ją w Polsce po raz drugi, gdy wiele cenniejszych pozycji nie może doczekać ukazania się po polsku.

MAREK SOBCZYK

[1]) Dennie Van Tassel: Praktyka programowania. Tłum. z ang. St. Szpakowicz. WNT, Warszawa 1982, wyd. 2, nakł. 8000 egz., s. 300, cena 180 zł

„Tablice rozproszone"

Techniki wyszukiwania informacji cieszą się niesłabnącym zainteresowaniem informatyków od pierwszych praktycznych zastosowań komputerów. Już na początku lat pięćdziesiątych naszego stulecia Hans Peter Luhn pracując dla firmy IBM wpadł na pomysł zastosowania do rozwiązania tego zagadnienia tzw. funkcji mieszających (ang. *hash functions*). Jego metoda, zwana również listową (ang. *chaining*), zapoczątkowała burzliwy rozwój pewnej klasy algorytmów wyszukiwania informacji. Ich wspólną cechą jest zorganizowanie pamięci maszyny w tablice rozproszone (ang. *hash tables*). Podejście Luhna zrewolucjonizowało

sposób patrzenia informatyków na problemy organizacji baz danych, tablic symboli w kompilatorach itp., stwierdzono bowiem, że średnia liczba prób wykonywanych przy wyszukaniu danego klucza jest stała i nie zależy od liczby kluczy umieszczonych w tablicy.

Chociaż od opracowania i zbadania pierwszych metod obsługi tablic rozproszonych upłynęło już około trzydziestu lat, zainteresowanie tym problemem nie słabnie. Pojawiają się ciągle nowe algorytmy dostosowane do specyficznych sytuacji. Technika tablic rozproszonych polega na znalezieniu funkcji h odwzorowującej dane klucze w adresy. Podstawową trudność w uzyskaniu takiej funkcji stanowi fakt, że w ogromnej większości zastosowań zbiór możliwych kluczy jest znacznie liczniejszy niż zbiór możliwych adresów. W takim przypadku może się zdarzyć, że różnym kluczom odpowiada ten sam adres. Tę niekorzystną sytuację nazywamy kolizją. Tak więc w metodach obsługi tablic rozproszonych stykamy się z dwoma podstawowymi zagadnieniami:

- jak wybrać funkcję h
- jak reagować na ewentualne kolizje.

Zapotrzebowanie informatyków na książkę poglądową omawiającą najbardziej typowe i sprawdzone w praktyce sposoby rozwiązywania tych problemów jest w kraju duże. Próbę zaspokojenia tych potrzeb podjął w swojej książce [1]) Romuald Jagielski.

Pierwsza część książki zawiera przegląd podstawowych funkcji mieszających. W połączeniu z szóstym rozdziałem, w którym zamieszczone jest porównanie poszczególnych metod, część ta dość dobrze orientuje czytelnika w problemie wyboru właściwej funkcji dla różnych zbiorów kluczy. Na marginesie należy stwierdzić, że funkcje mieszające trafniej można by nazwać funkcjami rozpraszającymi — z tego względu, że wymagamy od nich, aby jak najlepiej „rozpraszały" klucze po całej tablicy.

W dalszych rozdziałach książki omawiane są najbardziej typowe metody usuwania kolizji oraz reorganizacji tablic rozproszonych. I tu od razu zwraca uwagę duża niedbałość autora w zapisie algorytmów. Weźmy dla przykładu algorytm *B*. Dla tej samej zmiennej używane są tu wymiennie dwa oznaczenia d_0 oraz d, a że niczym się one mają nie różnić — czytelnik musi się tego domyśleć sam. Dalej: autor używa wskaźnika *LINK* beztrosko odnosząc go zarówno do tablicy T (kroki 3—4) jak i do N (kroki 3,4,6), choć uprzednio definiował *LINK(d)* jako wskaźnik umieszczony w *T[d]*!

Uwagi powyższe mniej lub bardziej zachowują moc także w innych przypadkach, np. w stosunku do algorytmów *C*, *G*, *E*. Przyjrzyjmy się z kolei algorytmowi *O*. Występujące w nim pole *TAG* zdefiniowane zostało przez autora w ten sposób, że *TAG(i)* = *1*, gdy rekord zapisany w *i-tym* polu tablicy T zmienił już swą lokalizację. Co więc oznacza *TAG(x)* w szóstym kroku, gdzie x jest rekordem chwilowo nie związanym z tablicą? Tego można się wprawdzie domyśleć, ale może wygodniej byłoby dla czytelnika, gdyby autor używał dla zapisu algorytmów języka, w którym można opisywać dane w sposób strukturalny (np. PASCALA). Czytelnik nie musiałby zastanawiać się podczas lektury, co autor miał na myśli, a zapis miałby jednoznaczną semantykę.

A algorytm *Q* usuwania zapisów z tablicy z listami synonimów? Powinien być kompatybilny z algorytmem *L* wstawiania zapisów, ale jeśli ktoś spróbuje wykorzystać w praktyce *L* i *Q* razem, napotka nowy problem, gdyż z książki nie wynika jak po działaniu *Q* w algorytmie *L* w kroku *3.1* lub *4* efektywnie znaleźć „adres dowolnej pustej działki".

Podobne usterki sprawiają wrażenie, że książka pisana była w dużym pośpiechu. Odczucie to pogłębia fakt, iż w algorytmach trafiają się błędy logiczne. Prześledźmy algorytm *M*. Okazuje się, że nie ma on własności stopu. Zapętli się na przykład wtedy, gdy tablica jest całkowicie zapełniona, szukamy zaś klucza mniejszego niż dowolny inny klucz znajdujący się w tablicy.

Zastanawiające jest także, dlaczego autor zaprezentował algorytm Radkego, skoro już obok opisany jest algorytm Day'a będący znacznie uproszczoną, a zarazem bardziej efektywną wersją poprzedniego. Celem książki nie było chyba przedstawianie historii rozwoju metod usuwania kolizji. Zupełnie zbędne wydaje mi się także poświęcanie miejsca na opis kartotek rozproszonych w języku PL/I, natomiast brak jest tak ważnych algorytmów, jak np. modyfikacja Brenta. Moim zdaniem — książka zyskałaby na atrakcyjności, gdyby oprócz wybranych algorytmów prezentowana była matematyczna analiza ich złożoności.

Pomimo irytujących potknięć formalnych w prezentacji algorytmów, dla tych informatyków, którzy dotąd nie zetknęli się z problemem tablic rozproszonych, niewątpliwie ciekawe jest omówienie oraz porównanie poszczególnych metod i ich zastosowań. Po uważnym przeczytaniu książki czytelnik zorientowany będzie w podstawowych zagadnieniach związanych z tablicami rozproszonymi, a rozszerzeniu tej wiedzy chyba dobrze posłużą liczne przytoczone informacje bibliograficzne.

ANDRZEJ SZAŁAS

[1]) Romuald Jagielski: Tablice rozproszone. WNT, Warszawa 1982. Nakład 5000 egz., 116 str. cena 40 zł, seria BIO

Stały kontakt z INFORMATYKĄ gwarantuje tylko prenumerata

Zasady jej zamawiania — na czwartej stronie okładki

Kwestie terminologiczne

Zasadniczo jestem zwolennikiem wprowadzania obcych słów do języka polskiego, co wzbogaca naszą mowę przydając jej niuansów i ścisłości, czemu dawałem już wyraz m.in. w „Informatyce". I tak, popieram użycie słowa „record" pisanego przez „c", bowiem „zapis" ma znaczenie ogólniejsze, a „rekord" funkcjonuje w języku polskim od dawna w zupełnie innym sensie i nie ma co mnożyć homonimów.

Natomiast niekiedy wprowadza się jakieś słowo lub pojęcie w oparciu o niewłaściwe rozumienie jego sensu w języku oryginału. Sporo szkód przyniosło np. przetłumaczenie, w swoim czasie, określenia „management information system" wypaczające sens oryginału, jako „system informowania kierownictwa". Również na zasadzie nieporozumienia, angielskie słowo handicap (przeszkoda) funkcjonuje w polszczyźnie jako „ułatwienie".

Obecnie na każdym kroku używa się słowa „specyfikacja", gdy w angielskim tekście występuje „specification" (np. w artykułach „Terminologia języka ADA", „O języku służącym do formułowania specyfikacji" — „Informatyka" 6/82). Tymczasem angielski rzeczownik „specification" znaczy opis techniczny urządzeń. Nie ma zatem wiele sensu używanie określeń „specyfikacja kontekstu" (context specification) czy „język specyfikacji systemu informatycznego" (system specification language zamiast „opis kontekstu" i „opis języka systemu", przy czym niejasność i pretensjonalność takiego stylu jest dość oczywista.

ANDRZEJ KAMIEŃSKI
Warszawa

PS. A propos: Computers in the neighbourhood (informacja o IFIP'83) to raczej „Komputery w osiedlu", a nie „Komputery w stosunkach sąsiedzkich".

Dziękuję Autorowi listu za wnikliwe uwagi. Trudno odmówić mu racji, gdy twierdzi, że specyfikacja znaczy wykaz, a specification — opis techniczny, choć można również podać inne znaczenia podstawowe obu słów i o wiele więcej znaczeń pokrewnych. Informatyków powinno jednak mniej interesować, co znaczy w ogóle słowo specification, a bardziej — w jakim znaczeniu zostało użyte.

Według podręcznika języka ADA (lipiec 1980), context specification poprzedzający jednostkę kompilacji określa (ang. defines) inne jednostki kompilacji, od których zależy dana jednostka. Natomiast, według normy IEEE Std 729, specification jest to dokument, który opisuje w sposób pełny, ścisły i możliwy do sprawdzenia, wymagania, projekt, zachowanie lub inne właściwości systemu (komputerowego, przyp. J.Z.) lub jego części.

W takich właśnie, różnych znaczeniach użyto słowa specyfikacja w cytowanych artykułach, a więc nie jest to tylko opis kontekstu (context description) i opis systemu (system description). W obu przypadkach potocznym odpowiednikiem słowa specification może być słowo opis. Należy jednak pamiętać, że używamy języka technicznego i specification jest terminem, a więc jego polski odpowiednik musi być precyzyjny. Tego warunku nie spełnia słowo opis, gdyż w obu definicjach specification ma znamiona opisu formalnego. Jak zwykle bywa w praktyce, postąpiono najprościej — spolszczając nazwę angielską i nadano słowu specyfikacja nowe znaczenie — opis formalny.

Jestem przekonany, że kryterium zrozumiałości zostało tu spełnione (czy, jak chce Autor listu — zachowano jasność). Natomiast pretensjonalność wyrażania się jest cechą mającą charakter emocjonalny, dlatego trudno wypowiadać się o niej jednoznacznie. To co dla jednego jest pretensjonalne, dla innego może uchodzić nawet za przejaw skromności.

JANUSZ ZALEWSKI

GRY

Firma

Kolejną z prezentowanych w naszym cyklu gier komputerowych jest „Firma" — gra typu business. Celem gry jest osiągnięcie jak największych zysków finansowych, na pewnym wyimaginowanym rynku. Ze względu na ograniczone rozmiary rubryki prezentujemy najprostszą wersję gry.

Założenia gry:

- na rynku występują elementy i produkty
- produkt otrzymywany jest po przetworzeniu surowca w fabryce
- gra podzielona jest na 24 etapy (można je nazwać np. miesiącami)
- w każdym etapie gracz może dokonać dowolną liczbę operacji, wybieranych z poniższej listy
  — kupowanie albo sprzedawanie elementów na rynku (po aktualnej cenie)
  — sprzedawanie produktów
  — przesyłanie elementów do fabryk
  — budowanie fabryki
  — zakończenie etapu.

Każda rozgrywka rozpoczyna się od identycznej sytuacji: gracz dysponuje sumą 50 tys. $ i jedną fabryką, cena elementu na rynku wynosi 1000 $, produktu — 3000 $, fabryki — 90 tys. $.

Na początku każdego etapu wszystkie ceny są modyfikowane przy wykorzystaniu generatora liczb losowych (zmiany od —10% do +15% dotychczasowej ceny). Gotówka wzrasta o 0,6%.

Każde przesłanie surowców do fabryki związane jest z kosztem transportu (1000 $). Transportowane elementy mogą ulegać zniszczeniu (0—33%). W fabryce może być przetwarzanych jednocześnie najwyżej 80 elementów, przy czym w jednym etapie można przesłać do jednej fabryki nie więcej niż 40 elementów. Przetwarzanie elementu (aż do uzyskania produktu) trwa trzy etapy (element dostarczony do fabryki w etapie n pojawi się jako produkt w etapie n+3).

Budowanie fabryki polega na jednorazowym wpłaceniu aktualnej ceny. Budowa trwa sześć etapów.

W rozliczeniu gry istotne są wyłącznie posiadane zasoby finansowe (tzn. nie jest istotna liczba fabryk i znajdujących się w nich nieprzetworzonych elementów).

Mimo bardzo uproszczonych założeń, można przekonać się podczas gry, że wymaga ona tzw. ‚głowy do interesów'.

Maksymalne nasze dotychczasowe osiągnięcie — 15 363 925 $.

Z. ŚWIRSKI

■ W dniach 18—21 października br. w Budapeszcie (Węgry) odbędzie się trzecie międzynarodowe sympozjum na temat zastosowań mikrokomputerów i mikroprocesorów μP'83 przygotowane przez Naukowe Towarzystwo Telekomunikacyjne, Towarzystwo Informatyczne im. Johna v. Neumana oraz Naukowe Towarzystwo Pomiarów i Automatyki przy współudziale Węgierskiej Akademii Nauk i Ministerstwa Przemysłu. Organizatorzy nie ograniczają zakresu prezentowanych prac i oczekują oryginalnych referatów związanych z zastosowaniami mikroinformatyki.

*

■ Brytyjska firma INMOS PLC, która zamierza w 1984 wypuścić na rynek mikroprocesor o nazwie TRANSPUTER, rozpoczęła produkcję oprogramowania związanego z nową maszyną. W pierwszym etapie opracowano nowy język, ukierunkowany na operacje współbieżne i możliwy do realizacji zarówno na TRANSPUTERZE, jak i innych obecnie produkowanych mikroprocesorach. Język ten powstał pod wyraźnym wpływem prac nad komunikującymi się procesorami sekwencyjnymi, prowadzonych na Uniwersytecie Oksfordzkim przez C.R. „Tony" Hoare'a. Głównym celem firmy INMOS jest opracowanie sprzętu pozwalającego w sposób efektywny implementować języki ze współbieżnością. W związku z tym każdy współdziałający proces będzie miał swój procesor z własną pamięcią. Oznacza to, że w jednym układzie będzie zawartych kilka procesorów i że zapewni się im sprawną wzajemną komunikację. Wyprzedzające prace nad oprogramowaniem mają stworzyć rynek dla nowego wyrobu, „oswajając" przyszłych klientów z tym typem maszyny.

*

■ W dniach 1—4 lutego br. obradowało w Bukareszcie (Rumunia) doroczne siódme posiedzenie Rady do spraw Zastosowań Środków Techniki Obliczeniowej Międzyrządowej Komisji ds. Elektronicznej Techniki Obliczeniowej (MKETO jest organizacją współpracy europejskich krajów socjalistycznych i Kuby). Głównym zadaniem było ustalenie tematów wspólnych prac podejmowanych w dziedzinie technologii i standaryzacji oprogramowania, systemów programów inżynierskich oraz wspomagających nauczanie i zarządzanie. Dotychczas, niezależnie od tego jaki kraj wykonał system zgłoszony do planu współpracy, wszyscy pozostali członkowie Rady stawali się automatycznie jego współwłaścicielami. Ponieważ nie sprzyjało to rozwojowi oprogramowania, postanowiono (i był to najważniejszy wynik tych obrad) zalecić prowadzenie dalszych prac na zasadach kontraktowych. Przy tej metodzie wszelkie prawa do rozpowszechniania produktów i korzyści handlowych należą do autorów. Przewiduje się wystawianie na sprzedaż zarówno oprogramowania wykonanego w jednym kraju, jak i powstałego w wyniku wspólnego działania kilku partnerów. M. in. stronie bułgarskiej powierzono sformułowanie zasad określania praw do programu na podstawie wkładu pracy poszczególnych partnerów.

■ Pod egidą rządowej Agencji Nauki i Techniki rozpoczęto w Japonii realizację trzyletniego programu, który ma doprowadzić do stworzenia systemu automatycznego tłumaczenia tekstów technicznych. Głównym celem przedsięwzięcia jest tłumaczenie japońskich prac na angielski (do światowego rozpowszechniania) oraz napływających z zewnątrz informacji w języku angielskim — na japoński, przede wszystkim dla personelu zarządzającego badaniami (większość badaczy czyta te prace w oryginale). System zostanie zaprogramowany w języku LISP i zrealizowany na amerykańskim komputerze zaprojektowanym specjalnie do pracy w tym języku. Do podstawowego słownika — opracowanego przez rządowe Laboratorium Elektrotechniczne — Japońskie Centrum Informacyjne Nauki i Techniki doda ok. 700 tys. terminów specjalistycznych. Przewidywany koszt programu: 2—3,2 mln dol.

*

■ Produkcja mikrokomputerów — rozpoczęta kiedyś przez małe niezależne firmy — znalazła się również w orbicie zainteresowań przedsiębiorstw od dawna produkujących duże maszyny cyfrowe. Po 16 miesiącach od wprowadzenia na rynek amerykański, IBM Personal Computer doczekał się swojej europejskiej premiery. Jest to wersja nieco zmodyfikowana — zastąpiono układy RAM 16 K układami 64 K, co spowodowało, że obecnie IBM PC dysponuje pamięcią od 64 do 512 K bajtów (zaledwie połowa możliwości adresowania użytego procesora INTEL 8038). W miejsce jednostronnie zapisywanych dyskietek o średnicy 5 1/4" (ok. 133 mm) wprowadzono ich zapis obustronny, co zwiększyło pojemność do 320 K bajtów. Ten nieco opóźniony europejski start spowodował, że maszyna ta została poprawiona nie tylko z punktu widzenia parametrów technicznych. Obecnie użytkownik w Europie ma szeroki wybór gotowego oprogramowania, a także stosowania urządzeń pomocniczych, produkowanych poza koncernem IBM.

*

■ Firma DISER (USA) wprowadziła na rynek komputer o nazwie LILITH, skonstruowany przez profesora Niklausa Wirtha i jego ekipę z politechniki federalnej w Zurychu (Szwajcaria). Tę małą maszynę przeznaczono przede wszystkim do zastosowań w dziedzinie projektowania wspomaganego komputerem (CAD) i redagowania dokumentów. Podstawowym językiem LILITH jest oczywiście stworzona przez profesora Wirtha MODULA 2, do której dopasowano architekturę sprzętu. Sercem systemu jest mikroprocesor segmentowy AM 2901, wielowejściowa pamięć operacyjna o pojemności 128 K słów 16-bitowych oraz pamięć mikroprogramów o pojemności 2 K. LILITH ma także dysk kasetowy o pojemności 10 M bajtów, interfejs szeregowy V-24 (RS-232), klawiaturę i ekran o wysokiej rozdzielczości (549 × 768 punktów), przyłączony przez magistralę 64-bitową, co zapewnia jego szybkie odświeżanie. Ze względu na wagę zastosowań graficznych, sterowanie obrazami odbywa się nie tylko za pomocą klawiatury, ale także specjalnego manipulatora (tzw. myszki) do bezpośredniego modyfikowania obrazu przez operatora.

*

■ Ciągle rozszerza się zakres maszyn używających systemu operacyjnego UNIX opracowanego przez Bell Laboratories. Firma HEWLETT-PACKARD nie tylko wybrała go do swojego nowego modelu 9000, ale zapowiedziała zastosowanie w szeregu innych wyrobach. Obecnie większość 16-bitowych, zbudowanych na bazie MC 68000 komputerów używa UNIXA — można go również znaleźć w oprogramowaniu podstawowym maszyn NCR, GOULD-SEL, AMDAHL, HONYWELL i UNIVAC. Według Jean Yates, redaktora publikacji firmowej „Unix newsletter", w ciągu roku do tego towarzystwa powinny dołączyć firmy BURROUGHS, IBM, TEXAS INSTRUMENTS, NIXDORF i SIEMENS, co spowoduje, że będzie to „jeden standardowy system operacyjny w całej gamie sprzętu — od mikrokomputerów do największych maszyn". Dodajmy jeszcze, że autorzy UNIXA — Dennis M. Ritchie i Ken Tomphson — otrzymali w 1982 doroczną nagrodę amerykańskiego czasopisma ELECTRONICS. Należy podkreślić, że po raz pierwszy w dziewięcioletniej historii przyznano ją za osiągnięcia w dziedzinie oprogramowania.

*

■ SIEMENS AG rozpoczyna w drugim kwartale br. dostawy 16-bitowego mikroprocesora SAB 80199, który jest pierwszym na świecie układem spełniającym wymagania pracy w czasie rzeczywistym. Będzie on szybszy o rząd wielkości od dotychczasowych kontrukcji, przy zachowaniu zgodności z magistralą 8086/88. Mogąc realizować do ośmiu zadań w trybie quasi-równoległym, SAB 80199 jest przeznaczony do sterowania szybkimi drukarkami, silnikami samochodowymi, a także w komputerach do sterowania procesami technologicznymi oraz do zastosowań telekomunikacyjnych.

*

■ Uniwersytet Carnegie-Mellon (USA) ma się wkrótce stać pierwszą uczelnią posiadającą ujednolicony sprzęt komputerowy do użytku studentów. Oczekuje się rychłego podpisania z IBM albo DIGITAL EQUIPMENT CORP. kontraktu na zainstalowanie sieci 8000 komputerów osobistych w całym miasteczku uniwersyteckim w Pittsburgu. Do czasu całkowitego zakończenia tego przedsięwzięcia planowanego na rok 1986, studenci obowiązani są dostarczyć własne komputery, tak jak obecnie muszą kupować książki, (przy czym opłata za ten sprzęt będzie rozłożona na cały okres studiów — cztery lata). Po zakończeniu nauki absolwenci będą mogli zabrać swoje komputery. Uzyskają też możliwość zdalnego dostępu do systemu Uniwersytetu Carnegie-Mellon z dowolnego miejsca Stanów Zjednoczonych.

Iszkowski W., Maniecki M.: Standaryzacja języka BASIC (1). Norma języka minimalnego

INFORMATYKA 1983, nr 5, s. 8

Charakterystyka zaakceptowanej przez ISO normy minimalnego podzbioru języka BASIC. Omówiono poszczególne elementy tej wersji języka: tekst programu, funkcje numeryczne, instrukcje sterujące, instrukcję iteracyjną, operacje napisowe oraz instrukcje wprowadzania i wyprowadzania.

Ишковски В., Манецки М.: Стандартизация языка BASIC (1). Стандарт минимального языка

INFORMATYKA 1983, № 5, стр. 8

Характеристика принятого ISO стандарта минимального подмножества языка BASIC. Обсуждены отдельные элементы этой версии языка: текст программы, числовые функции, инструкции управления, инструкция повторения, текстовые операции и инструкции ввода/вывода.

Oktaba H.: Klasy w LOGLANIE

INFORMATYKA 1983, nr 5, s. 12

Charakterystyka klas jako podstawowych modułów konstrukcji języka LOGLAN. Omówiono sposób deklarowania oraz obiekty klas, ilustrując je prostymi przykładami, a także korzyści wynikające z ich stosowania.

Октаба Г.: Классы в ЛОГЛАНЕ

INFORMATYKA 1983, № 5, стр. 12

Характеристика классов как основных модульев конструкции языка ЛОГЛАН. Обсуждено способ описания и объекты классов, поясняя их прямыми примерами, а также выгоды вытекающие из их применяния.

Owczarczyk J.: Parametry użytkowe wybranych systemów graficznych o organizacji rastrowej

INFORMATYKA 1983, nr 5, s. 15

Przegląd najpopularniejszych systemów graficznych o organizacji rastrowej produkcji amerykańskiej. Omówiono podstawowe parametry, typowe układy podstawowe, sprzęgi, układy specjalne, oprogramowanie oraz koszty tych systemów.

Овчарчык И.: Эксплуатационные параметры избранных графических систем растровой организации

INFORMATYKA 1983, № 5, стр. 15

Обзор наиболее популярных графических систем растровой организации американского производства. Обсуждены основные параметры, типичные основные схемы, интерфейсы, специальные схемы, программное обеспечение и стоимость этих систем.

Grabowski J.: Przegląd mikroprocesorów 16-bitowych (3). MOTOROLA 68000

INFORMATYKA 1983, nr 5, s. 18

Trzecia część charakterystyki najbardziej popularnych mikroprocesorów 16-bitowych, obejmująca omówienie mikroprocesora MOTOROLA 68000. Omówiono rejestry, tryby adresowania, listę rozkazów oraz przerwania programu.

Грабовски Й.: Обзор 16-битовых микропроцессоров (3). MOTOROLA 68000

INFORMATYKA 1983, № 5, стр. 18

Третья часть характеристики найболее популярных 16-битовых микропроцессоров содержащая обсуждение микропроцессора MOTOROLA 68000. Охарактеризованы регистры, режимы адресации, набор команд и прерывания программы.

Dawidowski J.: Komputer osobisty ZX81

INFORMATYKA 1983, nr 5, s. 21

Charakterystyka komputera osobistego firmy SINCLAIR, dostarczanego przez jedno z przedsiębiorstw polonijno-zagranicznych na rynek krajowy. Omówiono architekturę sprzętu, oprogramowanie, możliwości zastosowań oraz wyniki i ocenę próbnej eksploatacji.

Давидовски Й.: Персональный компьютер ZX81

INFORMATYKA 1983, № 5, стр. 21

Характеристика персонального компьютера фирмы SINCLAIR поставляемого одним из полонийно-заграничных предприятий на внутренний рынок. Обсуждена архитектура оборудования, программное обеспечение, возможности применений, результаты и оценка вступительной эксплуатации.

Furman M., Karpiński K., Nowicki Z., Solak J.: Rozszerzenie pakietu informacyjno-wyszukiwawczego SIM-400M

INFORMATYKA 1983, nr 5, s. 24

Charakterystyka zmian i rozszerzeń pakietu informacyjno-wyszukiwawczego SIM-400M, wykonanych w Instytucie Informacji Naukowej, Technicznej i Ekonomicznej w Warszawie. Omówiono zmiany dotyczące modułów wsadowych oraz nowe moduły konwersacyjne, a także kierunki prac zmierzających do zwiększenia możliwości funkcjonalnych pakietu.

Фурман М., Карпиньски К., Новицки З., Солак Й.: Расширение информационно-поискового пакета SIM-400M

INFORMATYKA 1983, № 5, стр. 24

Характеристика изменений и расширений информационно-поискового пакета SIM-400M, разработанных Институтом Научно-Технической и Экономической Информации в Варшаве. Обсуждено изменения касающиеся садочных модульев и новые разговорные модули, а также направления работ, имеющие в виду повышение функциональных возможностей пакета.

Rempała Z., Rygał R.: Translator języka adresów bezwzględnych dla procesora autonomicznego CAMAC typu 131

INFORMATYKA 1983, nr 5, s. 27

Charakterystyka translatora, obejmująca omówienie języka źródłowego, działanie translatora oraz możliwości jego eksploatacji.

Ремпала З., Рыгал Р.: Транслятор языка абсолютного для автономного процессора CAMAC типа 131

INFORMATYKA 1983, № 5, стр. 27

Характеристика транслятора, охватывающая оговорку языка исходного, деятельность транслятора и возможности его эксплуатации.

Iszkowski W., Maniecki M.: BASIC language standarization (1). Minimal BASIC standard

INFORMATYKA 1983, No. 5, p. 8

Characteristics of the ISO standard for BASIC minimal subset. Individual elements of this BASIC version: program's text, numeric statements, control statements, strings and I/O operations are discussed.

Iszkowski W., Maniecki M.: BASIC-Sprache Normierung (1). Minimal BASIC Norm

INFORMATYKA 1983, Nr. 5, S. 8

Eine Charakteristik der von ISO akzeptierten Norm für die minimale BASIC-Untermenge. Es wurden die einzelnen Elemente dieser Spracheversion: Programmtext, numerische Funktionen, Steuerbefehle, Iterationsbefehl, Aufschriftoperationen sowie Eingabe/Ausgabebefehle besprochen.

Oktaba H.: Classes in LOGLAN

INFORMATYKA 1983, No. 5, p. 12

Characteristics of classes as basic modules of the LOGLAN language structure. The way of declarating and classes objects, illustrated with simple examples, as well as advantages of their application, are discussed.

Oktaba H.: Klassen in LOGLAN

INFORMATYKA 1983, Nr. 5, S. 12

Eine Charakteristik von Klassen als Grundmodule der LOGLAN-Sprachestruktur. Es wurden Deklarierungsweise und Klassenobjekte, illustriert mit einfachen Beispielen, sowie Vorteile ihrer Anwendung, besprochen.

Owczarczyk J.: Application parameters of selected graphical systems with raster organisation

INFORMATYKA 1983, No. 5, p. 15

Review of the most popular US-made graphical systems with raster organisation. Main parameters, basic circuits, interfaces, special circuits, software and costs of these systems are discussed.

Owczarczyk J.: Gebrauchsparameter von ausgewählten graphischen Systeme mit Rasterorganisation

INFORMATYKA 1983, Nr. 5, S. 15

Eine Vorschau der populärsten graphischen Systeme mit Rasterorganisation amerikanischer Herkunft. Es wurden Basjsparameter, typische Standardschaltungen, Schnittstellen, Sonderschaltungen, Software, sowie die Kosten dieser Systeme, besprochen.

Grabowski J.: 16-bit microprocessor survey (3). MOTOROLA 68000

INFORMATYKA 1983, No. 5, p. 18

Third part of the most popular 16-bit microprocessors characteristics — presentation of MOTOROLA 68000 microprocessor. Registers, addressing modes, instruction set and program interrupts are discussed.

Grabowski J.: Vorschau von 16-Bits-Mikroprocessoren (3). MOTOROLA 68000

INFORMATYKA 1983, Nr. 5, S. 18

Dritter Teil der Charakteristik der populärsten Vertreter von 16-Bits Mikroprocessoren, die eine Besprechung des MOTOROLA 68000 Mikroprocessors umfasst. Es wurden Register, Adressierungsmodi, Befehlsliste und Programmunterbrechungen besprochen.

Dawidowski J.: Personal computer ZX81

INFORMATYKA 19833, No. 5, p. 21

Characteristics of SINCLAIR's personal computer, supplied on polish market by one of polish-foreign enterprises. Hardware architecture, software, application possibilities, as well as results and evaluation of the experimental operation are discussed.

Dawidowski J.: Personalrechner ZX81

INFORMATYKA 1983, Nr. 5, S. 21

Eine Charakteristik des Personalrechners von SINCLAIR, der auf den polnischen Markt von einem polnisch-ausländischen Unternehmen geliefert wird. Es wurden Hardwarearchitektur, Software, Anwendungsmöglichkeiten sowie Ergebnisse und Beurteilung des Probebetriebes besprochen.

Furman M., Karpiński K., Nowicki Z., Solak J.: Extension of the SIM-400M information retrieval package

INFORMATYKA 1983, No. 5, p. 24

Characteristics of the SIM-400M information retrieval package, changes and extensions elaborated in the Institute of Scientific, Technical and Economic Information in Warsaw. Changes in batch modules and new interactive modules, as well as directions for future extensions of the package functional possibilities are discussed.

Furman M., Karpiński K., Nowicki Z., Solak J.: Erweiterung des SIM—400M Recherchepakets

INFORMATYKA 1983, Nr. 5, S. 24

Eine Charakteristik von Veränderungen und Erweiterungen des SIM-400M Recherchepakets, die im Institut für Wissenschaftliche, Technische und Ekonomische Information in Warschau erarbeitet wurden. Es wurden Veränderungen in Stappelmodulen und zwei neue Konversationsmodule, sowie Arbeitsrichtungen, die zur Steigerung der Paketsmöglichkeiten führen werden, besprochen.

Rempała Z., Rygał R.: Absolute address language translator for CAMAC autonomic processor type 131

INFORMATYKA 1983, No. 5, p. 27

Translator characteristics containing presentation of the source language, translators operation and its application possibiblities.

Rempała Z., Rygał R.: Übersetzungsprogramm der absoluten Sprache für den autonomischen CAMAC-Prozessor von Typ 131

INFORMATYKA 1983, Nr. 5, S. 27

Eine Charakteristik des Übersetzungsprogramms, die Besprechung der Quellsprache, Arbeitsweise des Übersetzers und seine Betriebsmöglichkeiten umfasst.

Moje podglądy

Gdyby red. Marek Rostocki z Przeglądu Technicznego zorganizował spotkanie ze zdobywcą tytułu Mistrza Techniki 1982, to daję sobie obciąć prawą rękę, jeśli pierwsze pytanie nie brzmiałoby tak: „Czy Panowie nie wstydzą się brać nagrody za to, co Amerykanie zrobili już 10 lat temu?". Mierząc czas nie w latach, ale wzrostem produkcji, spadkiem cen lub wielkościami technicznymi, jak gęstość upakowania elementów lub szerokość ścieżek w mikroukładzie, łatwo stwierdzić, że 10 lat w elektronice to cały wiek. A więc Amerykanie wynaleźli mikroprocesor — elektroniczny wiek temu.

Gdy wiek mija w ciągu dziesięciu lat trudno to zauważyć, także w wymiarze społecznym, nie mając podglądu. Tak jak w telewizji — trzeba zobaczyć „z boku". Czy zabrakło podglądu komsji przydziału nagród, choć opierała się na opiniach specjalistów? Nie wiem, co by odpowiedzieli laureaci na postawione na wstępie pytanie, ale należy odpowiedzieć szczerze przykładając właściwą miarę — w skali krajowej to osiągnięcie liczy się bardzo, bo jest jedyne i stwarza dalsze perspektywy, niemniej jest kroplą w morzu elektroniki światowej. Czy będzie kamieniem milowym w rozwoju techniki krajowej — to dopiero się okaże, gdyż za tym wyzwaniem musi iść atak frontalny.

Podglądu zabrakło też prof. Adamowi Schaffowi, który twierdzi, że „jest dziedzina, w której Polska doskonale może prosperować — produkcja tzw. software'u czyli programów dla komputerów". Czy jest to możliwe? Ano, spójrzmy na podgląd: jak pracuje się przy produkcji oprogramowania?

W Polsce mało kto wie jak to robić efektywnie, lekceważone są bowiem stosowane już powszechnie na całym świecie reguły przemysłowej produkcji oprogramowania. Kto w Polsce był członkiem zespołu zwanego „chief programmer team", powołanego do produkcji oprogramowania? Czy słyszał ktoś o przedsięwzięciu krajowym, które realizowałoby zasadę „IV and V" (ang. independent verification and validafion), która polega na powierzeniu jednemu wykonawcy wytworzenia oprogramowania, a drugiemu kontrahentowi — oceny fachowej i przyjęcia tego produktu? Są to metody stosowane już od dawna, ale nie w Polsce.

Inną sprawą są zagadnienia tzw. środowiska programowego, tj. oprogramowania podstawowego, mającego charakter narzędzi, które muszą być dane, np. do wytworzenia oprogramowania użytkowego według wymagań konkretnego kontrahenta. Moim zdaniem jest pewne, że z braku takiego środowiska upadnie przedsięwzięcie pn. LOGLAN. Mam ogromny szacunek dla jego autorów i realizatorów, ale na upowszechnienie tego bardzo nowoczesnego języka nie mają co liczyć, dopóki nie powstanie dla niego środowisko programowe, a tego trudno się spodziewać. W tej dziedzinie podobnie jak w mikroelektronice, grożą nam straty nie do odrobienia. Wraz z rozwojem prac nad językiem ADA i zablokowaniem transferu tych osiągnięć w naszą stronę powstanie zupełnie nowa sytuacja — już wkrótce w krajach rozwiniętych technologicznie dużą część oprogramowania — czy chcemy, czy nie — będzie pisać się w języku ADA. Węgrzy zgrupowali kilkudziesięciu najlepszych, młodych programistów z różnych instytutów i nie oglądając się na nikogo ruszyli do prac nad ADĄ. Tak jak dziś każdy nabywca pyta, czy urządzenie zawiera mikroprocesor, jutro będzie pytać, czy jest programowane w ADZIE, inaczej go nie kupi. Pozbawimy się wielu możliwości, jeśli w porę nie podejmiemy odpowiednich prac.

Jakie jednak mamy warunki do podjęcia tych prac? Doszliśmy zatem do pytania o infrastrukturę. Gdzie są komputery, na których mamy wytwarzać to oprogramowanie, gdzie jest serwis, gdzie są szkolenia dla początkujących i zaawansowanych? Nie będę podawał przykładów, bo każdy z nas zna wiele. Jedynie do przesuwania liczydeł nie potrzeba narzędzi.

Oprócz metod, organizacji i infrastruktury jest jeszcze motywacja do pracy. Kadra wykształconych specjalistów — to za mało. Zamiast rozwodzić się nad tym tematem, posłużę się autorefleksją — gdybym miał motywację do pracy w zawodzie, to nie byłbym kierownikiem ZOINTE. Zresztą, wykształcenia też nam zacznie wkrótce brakować. Na Węgrzech w każdej szkole podobno są już 2—3 mikrokomputery. A bezpośrednie kontakty międzynarodowe, gdzie przepływ informacji technicznych jest najszybszy?

To wszystko widzę na podglądzie. Teraz już wiem, za co można lub trzeba przyznawać ludziom nagrody. Gołymi rękami można postawić piec lub ogrodzić pole, ale nie tworzyć dziedziny, na której może się oprzeć rozwinięty gospodarczo kraj. Kraje, które zdobyły czołową pozycję międzynarodową dzięki rozwojowi technologii, zawdzięczają ją znakomitej organizacji scalającej wszystkie elementy niezbędne do tego rozwoju. Prof. Andrzej Kobus wraz z zespołem powinien otrzymać tę nagrodę nie tylko za osiągnięcie techniczne, ale także za umiejętność zorganizowania pracy w warunkach przypominających często walkę z wiatrakami.

Toteż, gdy prof. Tadeusz Porębski mówił w niedawnym wywiadzie, że „nie stać nas na wielki postęp w elektronice", to należy mu wierzyć, bo to prawda. Ale nie dlatego, że rzeczywiście brak nam pieniędzy. Nie stać nas raczej na opłacanie bałaganu i topienie pieniędzy w tragicznej otchłani antyorganizacji.

A może nie stać nas na mądrość zainwestowania w elektronikę i informatykę, bo nie wiemy jak przełamać bariery? Wystarczy zerknąć na podgląd.

JANUSZ ZALEWSKI

Zasady prenumeraty

Zamówienia i przedpłaty na prenumeratę INFORMATYKI przyjmuje: Wydawnictwo SIGMA, skrytka 1004, 00-950 Warszawa; konto bankowe: nr 1036-7490-139-11 III O/M NBP Warszawa.

Instytucje przesyłają zamówienia, zawierające: liczbę zamawianych egzemplarzy, okres prenumeraty, pełny adres zamawiającego, adresy odbiorców oraz numer konta bankowego zamawiającego. Dopisek PRENUMERATA STAŁA przy zamówieniu całorocznym oznacza, że zamawiający nie musi już ponawiać zamówień, lecz jedynie dokonywać corocznie przedpłaty wg obowiązujących cen.

Prenumeratorzy indywidualni dokonują przedpłaty na ww. konto, podając swój dokładny adres z kodem pocztowym oraz imię i nazwisko, a na odwrocie odcinka dla adresata — posiadacza rachunku — tytuł czasopisma, liczbę egzemplarzy oraz okres prenumeraty.

Terminy przedpłat:
do 25 listopada — I kwartał, I półrocze, cały następny rok (w tym prenumerata stała);
do 10 marca — II kwartał;
do 10 czerwca — III kwartał, II półrocze;
do 10 września — IV kwartał.

Prenumerata kwartalna — 225 zł, półroczna — 450 zł, roczna — 900 zł. Prenumerata ze zleceniem wysyłki za granicę jest dwukrotnie droższa.

Od 1 lipca br. wprowadzona zostaje prenumerata ulgowa dla członków stowarzyszeń naukowo-technicznych NOT, studentów wyższych uczelni, uczniów szkół zawodowych. Warunkiem jej uzyskania jest poświadczenie blankietu dla nabywcy indywidualnego na odcinku dla adresata przez właściwe stowarzyszenie NOT, wyższą uczelnię lub szkołę zawodową. Cena prenumeraty ulgowej: kwartalnie — 150 zł, półrocznie — 300 zł, rocznie — 600 zł.

Dodatkowych informacji udziela:
Dział Handlowy Wydawnictwa SIGMA, Warszawa, ul. Bartycka 20, p. 14; tel. 40-37-31 lub 40-00-21 wew. 207.

W numerze 6/83 przedstawimy:

Huzar A.: Programowanie współbieżne w systemie OS/IS na poziomie PASCALA

Grabowski J.: Przegląd mikroprocesów 16-bitowych (4). TEXAS INSTRUMENTS 9900

Iszkowski W., Maniecki M.: Standaryzacja języka BASIC (2). Moduł rdzeniowy języka

Owczarczyk J.: Projektowanie systemu graficznego. Część 1

Oktaba H.: Prefiksowanie klasami w LOGLANIE

W najbliższych numerach:

- Pamięci dyskowe typu Winchester
- Programowanie za pomocą współprogramów
- Automatyczne redagowanie tekstów
- Dyskowy symulator ODRA-RIAD
- CESARO — system analizy i rozpoznawania obrazów wizualnych
- GOBI — system do obliczeń geometrycznych
- Wielkie wydarzenia w rozwoju oprogramowania